

# **Customized Topology Optimization for Additive Manufacturing**

by

Davin Jankovics

A thesis submitted to the  
School of Graduate and Postdoctoral Studies in partial  
fulfillment of the requirements for the degree of

**Master of Applied Science in Mechanical Engineering**

The Faculty of Engineering and Applied Science

Ontario Tech University

Oshawa, Ontario, Canada

August 2019

© Davin Jankovics, 2019

# Thesis Examination Information

Submitted by: Davin Jankovics

Master of Applied Science in Mechanical Engineering

Thesis title: Customized Topology Optimization for Additive Manufacturing
---

An oral defense of this thesis took place on August 9, 2019 in front of the following examining committee:

## Examining Committee:

Chair of Examining Committee	Dr. Xianke Lin
Research Supervisor	Dr. Ahmad Barari
Examining Committee Member	Dr. Martin Agelin-Chaab
Thesis Examiner	Dr. Jana Abou Ziki

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

## Abstract

One of the biggest limitations of additive manufacturing (AM) is the resulting production times. Due to the layer-based method of material deposition, the time to produce a single part is substantial compared to techniques like injection molding or casting. However, the level of part complexity that can be achieved using AM processes is also unrivaled. This is a perfect match for the structural design method of topology optimization. It often produces parts with complex organic features that can perform substantially better in terms of weight and stiffness compared to their conventionally designed counterparts. Thus, an AM topology optimization constraint is developed to address the limitations of these processes while maintaining the advantages of the optimization. This is achieved through a penalization scheme applied to boundary contours identified through a slicing mechanism. The result is parts that print substantially faster, while only losing some stiffness compared to the normal topology optimization.

**Keywords:** Topology Optimization; Additive Manufacturing; Print Time Reduction; Slicing; Finite Element Analysis

## **Author's Declaration**

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Davin Jankovics

---

## Statement of Contributions

Part of the work described in Chapter 4, and small portions of other chapters have been published as:

- D. Jankovics, H. Gohari, M. Tayefeh, and A. Barari, “Developing Topology Optimization with Additive Manufacturing Constraints in ANSYS®,” in *16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018*, 2018, vol. 51, no. 11, pp. 1359–1364.
- D. Jankovics, H. Gohari, and A. Barari, “Constrained Topology Optimization for Additive Manufacturing of Structural Components in Ansys®,” in *Proceedings of The Canadian Society for Mechanical Engineering International Congress 2018*, 2018.
- D. Jankovics and A. Barari, “Customization of Automotive Structural Components using Additive Manufacturing and Topology Optimization,” in *13th IFAC Workshop on Intelligent Manufacturing Systems*, (Pre-print), 2019.

Part of the work described in Chapter 3 & 5 are being prepared for publication as:

- D. Jankovics and A. Barari, “Topology Optimization for Reduction of Additive Manufacturing Print Time,” (In-preparation), 2019.

For all publications I performed the majority of the literature review, the methodology, results, and writing of the manuscript. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

## Acknowledgments

Here, I would like to give a little thanks to those that made this thesis possible. First, to my supervisor Dr. Ahmad Barari; your guidance and patience during the length of my undergraduate 4th year capstone project and masters was immeasurable. His belief in my capabilities at times far surpassed what I thought I could achieve, pushing me to new heights.

To my lab mates and colleagues: Cody Berry, Hossein Gohari, and Dylan Bender; your inputs and discussions were invaluable to both my success and enjoyment of my time in the lab. I can only hope I repaid even a small fraction of what I received. To all my friends and colleagues both in and outside of academics, thank you. I would also like to thank my examining committee, Dr. Jana Abou Ziki and Dr. Martin Agelin-Chaab for taking the time to evaluate my work.

To my dearest love, Marina Freire Welby; thank you for your unbelievably big heart and support over the past 6 years. Without your humour, positivity, patience, and love, I would never have made it this far. Coming home to you is truly the best part of my day. To my family, especially my mother and father, Diane and Dusan; your love and support do not go unnoticed. Only with your help could I have followed my dreams so closely.

# Table of Contents

Abstract .....	iii
Author's Declaration.....	iv
Statement of Contributions .....	v
Acknowledgments .....	vi
Table of Contents.....	vii
List of Tables.....	x
List of Figures.....	xi
Nomenclature .....	xvi
Introduction .....	1
1.1    Technical Background .....	3
1.1.1    Finite Element Analysis .....	3
1.1.2    Topology Optimization.....	4
1.1.3    Additive Manufacturing.....	8
Literature Review .....	10
2.1    Overview of Topology Optimization Methods.....	10
2.2    Manufacturing Constraints in Topology Optimization .....	11

2.3	Topology Optimization for Additive Manufacturing.....	14
2.4	Gaps in the Literature & Thesis Objectives.....	22
	Methodology .....	24
3.1	Reduction of the Perimeter .....	24
3.1.1	Ideal Perimeter .....	25
3.1.2	Perimeter Normalization.....	30
3.1.3	Mass Concentration Approach .....	32
3.2	Identification of Boundary/Perimeter Contours .....	34
3.2.1	Slicing .....	34
3.2.2	Ray-tracing Approach .....	35
3.3	Formation of the New Topology Optimization Constraint.....	36
3.3.1	Perimeter Constraint .....	37
3.3.2	New Topology Optimization Formulation.....	38
	Implementation.....	39
4.1	Implementation using Scripting & Commercial CAE Software .....	39
4.1.1	Justification for Utilizing Commercial FEA Code.....	40
4.1.2	Structure of the Program.....	42
4.2	Additive Manufacturing Perimeter Constraint .....	44
4.2.1	SIMP Implementation .....	44
4.2.2	Slicer Implementation.....	45
4.2.3	Perimeter Constraint Implementation.....	47
	Results & Validation.....	48

5.1	2D Cases .....	48
5.1.1	MBB Beam .....	49
5.1.2	Cantilever Beam .....	53
5.1.3	Multiple Load Case Bridge .....	56
5.2	3D Cases .....	59
5.2.1	3D MBB Beam .....	60
5.2.2	3D L-Bracket .....	65
5.3	Validation of Print Time Reduction .....	70
5.3.1	MBB Beam .....	71
5.3.2	Cantilever Beam .....	73
5.3.3	Bridge .....	76
5.3.4	3D MBB Beam .....	78
5.3.5	3D L-Bracket .....	81
5.4	Summary of Results .....	84
	Conclusion.....	87
	Bibliography.....	90

# List of Tables

<b>Table 5-1.</b> Percentage change of various quantities as compared to the base topology optimization with no AM perimeter penalty.....	85
---	----

# List of Figures

<b>Figure 1-1.</b> A typical FEA mesh. ....	4
<b>Figure 1-2.</b> General sequence of a topology optimization, with iterations shown. .	5
<b>Figure 1-3.</b> Checkerboard effect.....	7
<b>Figure 1-4.</b> FDM/FFF Process.....	8
<b>Figure 1-5.</b> Depiction of slicing an STL. ....	9
<b>Figure 1-6.</b> Various infill patterns from a popular open source slicer (Slic3r). ....	9
<b>Figure 2-1.</b> AM and topology optimization research papers over time (based on appearances in Google Scholar). ....	14
<b>Figure 2-2.</b> Staircase effect and cusp height in orange ( $\delta$ ). Black is the boundary of the original model, grey is the deposited material. ....	17
<b>Figure 3-1.</b> Print speed visualization of perimeters and infill (Generated by the open-source <i>Slic3r</i> ).....	25
<b>Figure 3-2.</b> Convex shape (left) and a non-convex shape (right).....	26
<b>Figure 3-3.</b> Process showing increase in perimeter when adding a cavity .....	27
<b>Figure 3-4.</b> Non-convex mirroring process.....	28

<b>Figure 3-5.</b> Regular polygons with 3, 4, 6 and 12 sides with approximately the same area.....	28
<b>Figure 3-6.</b> Plot showing how the number of edges of a polygon.....	31
<b>Figure 3-7.</b> Design domain in blue. Solid area is grey and white is void. Outer perimeters are shown in black, and inner perimeters in red. ....	33
<b>Figure 3-8.</b> Graphic of expanding perimeters. ....	34
<b>Figure 3-9.</b> Graphic of shrinking perimeters.....	34
<b>Figure 3-10.</b> Slicing procedure [61]. ....	35
<b>Figure 3-11.</b> Ray-tracing for inner/outer contour detection. Outer contours in are black, with inner contours in orange. Rays are in blue, with intersections marked as blue crosses.....	36
<b>Figure 4-1.</b> A simplified flowchart of the program interactions and logic.....	44
<b>Figure 4-2.</b> Example of meshes (1a & 2a) and their respective slices (1b & 2b). ....	46
<b>Figure 5-1.</b> 2D MBB beam setup. ....	49
<b>Figure 5-2.</b> Resulting geometry for the MBB beam. ....	50
<b>Figure 5-3.</b> Change in the MBB beam’s compliance and normalized perimeter based on AM perimeter penalty.....	51
<b>Figure 5-4.</b> Change in the MBB beam’s normalized perimeter over each iteration. ....	52
<b>Figure 5-5.</b> Change in the MBB beam’s compliance over each iteration. ....	53
<b>Figure 5-6.</b> 2D cantilever beam setup.....	53
<b>Figure 5-7.</b> Resulting geometry for the cantilever beam. ....	54

<b>Figure 5-8.</b> Change in the cantilever’s compliance and normalized perimeter based on AM perimeter penalty.....	55
<b>Figure 5-9.</b> Change in the cantilever’s normalized perimeter over each iteration.	55
<b>Figure 5-10.</b> Change in the cantilever’s compliance over each iteration. ....	56
<b>Figure 5-11.</b> 2D bridge setup.....	56
<b>Figure 5-12.</b> Resulting geometry for the bridge. ....	57
<b>Figure 5-13.</b> Change in the bridge’s compliance and normalized perimeter based on AM perimeter penalty.....	58
<b>Figure 5-14.</b> Change in the bridge’s normalized perimeter over each iteration. ..	58
<b>Figure 5-15.</b> Change in the bridge’s compliance over each iteration. ....	59
<b>Figure 5-16.</b> 3D MBB beam setup. ....	60
<b>Figure 5-17.</b> Part I - Resulting geometry for the 3D MBB beam.....	61
<b>Figure 5-18.</b> Part II - Resulting geometry for the 3D MBB beam.....	62
<b>Figure 5-19.</b> Change in the 3D MBB beam’s compliance and total normalized perimeter based on AM perimeter penalty. ....	63
<b>Figure 5-20.</b> Change in the 3D beam’s normalized perimeter over each iteration. ....	64
<b>Figure 5-21.</b> Change in the 3D beam’s compliance over each iteration. ....	64
<b>Figure 5-22.</b> 3D L-bracket beam setup.....	65
<b>Figure 5-23.</b> Part I - Resulting geometry for the 3D L-bracket.....	66
<b>Figure 5-24.</b> Part II - Resulting geometry for the 3D L-bracket.....	67
<b>Figure 5-25.</b> Change in the 3D L-bracket’s compliance and total normalized perimeter based on AM perimeter penalty. ....	68

<b>Figure 5-26.</b> Change in the 3D L-bracket’s total normalized perimeters over each iteration (for clarity only every 10 <sup>th</sup> result is plotted). .....	69
<b>Figure 5-27.</b> Change in the 3D L-bracket’s compliance over each iteration (for clarity only every 10th result is plotted). .....	69
<b>Figure 5-28.</b> Resulting toolpaths for various MBB beam results with identical slicer settings (Slic3r). .....	71
<b>Figure 5-29.</b> Change in the MBB beam’s simulated print time. ....	72
<b>Figure 5-30.</b> Change in the MBB beam’s simulated material usage. ....	72
<b>Figure 5-31.</b> Change in the MBB beam’s simulated print time and material usage based on the selected infill percentage for $pAM = 1.9$ . ....	73
<b>Figure 5-32.</b> Resulting toolpaths for various cantilever beam results with identical slicer settings (Slic3r). .....	74
<b>Figure 5-33.</b> Change in the cantilever beam’s simulated print time. ....	74
<b>Figure 5-34.</b> Change in the cantilever beam’s simulated material usage. ....	75
<b>Figure 5-35.</b> Change in the cantilever beam’s simulated print time and material usage based on the selected infill percentage for $pAM = 1.7$ . ....	75
<b>Figure 5-36.</b> Resulting toolpaths for various bridge results with identical slicer settings (Slic3r). .....	76
<b>Figure 5-37.</b> Change in the bridge’s simulated print time. ....	77
<b>Figure 5-38.</b> Change in the bridge’s simulated material usage. ....	77
<b>Figure 5-39.</b> Change in the bridge’s simulated print time and material usage based on the selected infill percentage for $pAM = 1.7$ . ....	78

<b>Figure 5-40.</b> Resulting toolpaths for various 3D MBB beam results with identical slicer settings (Slic3r).....	79
<b>Figure 5-41.</b> Change in the 3D MBB beam’s simulated print time. ....	80
<b>Figure 5-42.</b> Change in the 3D MBB beam’s simulated material usage. ....	80
<b>Figure 5-43.</b> Change in the 3D MBB beam’s simulated print time and material usage based on the selected infill percentage for $pAM = 1.9$ . ....	81
<b>Figure 5-44.</b> Resulting toolpaths for various 3D L-bracket results with identical slicer settings (Slic3r).....	82
<b>Figure 5-45.</b> Change in the 3D L-bracket’s simulated print time. ....	83
<b>Figure 5-46.</b> Change in the 3D L-bracket’s simulated material usage. ....	83
<b>Figure 5-47.</b> Change in the 3D L-bracket’s simulated print time and material usage based on the selected infill percentage for $pAM = 1.9$ . ....	84
<b>Figure 5-48.</b> Change in averaged quantities vs AM perimeter penalty.....	86

## Nomenclature

### Latin Letters

$a$	Length of polygon vertex to centroid (cm)
$A$	Cross-sectional area (cm <sup>2</sup> )
$b$	Length of polygon base (cm)
$c$	Compliance (J)
$e$	Element number
$f$	Volume fraction (cm <sup>3</sup> /cm <sup>3</sup> )
$F$	Global force vector
$h$	Height of polygon (cm)
$k_e$	Elemental stiffness matrix
$K$	Global stiffness matrix
$m$	Positive move limit
$n$	Number of edges
$N$	Number of elements
$p$	Penalty value
$P$	Perimeter (cm)
$P_N$	Normalized perimeter (cm/cm)
$r$	Radius of a circle (cm)
$S_T$	Total sensitivity
$u_e$	Elemental displacement vector
$U$	Global displacement vector
$V$	Material volume (cm <sup>3</sup> )
$V_0$	Design domain volume (cm <sup>3</sup> )

## Greek Letters

$\alpha$	Polygon apex angle ( $^{\circ}$ )
$\delta$	Cusp height (cm)
$\eta$	Numerical damping coefficient
$\theta$	Half of polygon apex angle ( $^{\circ}$ )
$\lambda$	Lagrangian multiplier
$\rho$	Element density
$\varphi$	Polygon angle ( $^{\circ}$ )

# Chapter 1

## Introduction

In many industries today, a successful product demands lightweight design, considerable structural performance, and cost-effective manufacturing. The challenge of meeting these requirements has led to the increased use of advanced engineering techniques during the design process. The availability of computational resources has also accelerated the use of techniques that rely on iterative numerical methods, ones that would be infeasible to do by hand. A major branch of these methods is known as structural optimization.

Within this discipline, three main groups exist; size, shape, and topology optimization. Size optimization focuses on determining the thicknesses of predefined members in a problem to satisfy a particular goal, such as minimizing deflection. Similarly, shape optimization modifies pre-existing geometric features to achieve similar goals. Topology optimization, in contrast, fully determines the resulting shape and features; in other words, the material distribution of a design domain. While these methods, and specifically topology optimization, produce lightweight designs

for a given set of boundary conditions, they are often difficult to manufacture using traditional means without extensive modifications.

Additive manufacturing (AM) is a method where material is deposited in layers to produce the final or near-final part geometry. It is also commonly referred to as 3D printing. Compared to conventional subtractive manufacturing methods such as milling and turning, AM methods bypass many restrictions in part geometry, making it an excellent match for producing topology optimized parts. However, AM is not being adopted to the extent that its strongest proponents would like to see. This is due in large part to lengthy production times. The purpose of this thesis then, will be to develop an additive manufacturing topology optimization constraint; which is designed to decrease the time it takes to produce a part while still gaining benefits from a structural optimization.

The structure of this thesis will be the following. First, a basic technical background of the most relevant topics will be detailed. Next, Chapter 2 will present a literature review for works related to topology optimization additive manufacturing constraints. Following this, Chapter 3 will describe the methods used to develop the new print time reduction constraint, followed by their implementation in Chapter 4. Then, Chapter 5 will cover validation of the constraint using multiple case examples. Finally, the impact of the new constraint and further improvements will be discussed in Chapter 6.

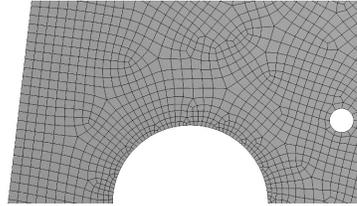
## 1.1 Technical Background

In general, topology optimization is a process in which a design domain is first defined and discretized into a mesh, a finite element analysis (FEA) is conducted, and then results from this are used to modify elements to favour structural stiffness. This process is then repeated in an iterative manner until the solution is considered complete. Due to the organic shapes that this process frequently produces, additive manufacturing is often the only suitable production method. However, AM methods have their own set of challenges and the exact process must be detailed to demonstrate the principle of a print time reduction scheme.

### 1.1.1 Finite Element Analysis

To gain a better understanding of how topology optimization achieves its goals, it is useful to introduce the basics of finite element analysis. For all but the simplest engineering problems, exact solutions cannot be easily obtained, if at all. This is generally due to the difficulty in solving the governing or boundary partial differential equations that make up the problem. To get past this, the solution can be approximated with numerical techniques, where a continuous design domain is broken up into discrete sub-domains, known as elements. These elements can take the form of various shapes and exist in what is known as a finite element mesh (seen in Figure 1-1). Based on the chosen shape and desired properties, each element is made of a varying number of nodes. These nodes generally lie at the vertices of the element, and along its edges. Based on their available degrees of freedom, nodes can be moved based on the connection to elements around them. Here, each element is

represented by a simple set of algebraic equations that relate the nodes degrees of freedom to each other. These are then combined into the full system of equations that models the problem.



**Figure 1-1.** A typical FEA mesh.

In a static structural FEA, each element can be represented by a linear spring, where based on its stiffness the element will “stretch” a certain distance when a force is applied. Elements can then be described numerically with what is known as a stiffness matrix, with the entire set of elements assembled into a global stiffness matrix. Boundary conditions, such as elements having certain degrees of freedom fixed, are then applied along with the forces on the model. Solving this global matrix, displacements of each element can be obtained, and from there quantities such as stress and strain can also be calculated. The main result that is of interest for topology optimization is an element’s strain energy density. This represents compliance, that is, the inverse of stiffness.

### 1.1.2 Topology Optimization

In general and for our purposes here, a topology optimization seeks to minimize compliance while meeting a certain design constraint such as a volume fraction. As described above, this relies first on describing a design domain and a set of boundary

conditions. This domain is then discretized via a finite element mesh. Based on the FEA results, the design domain is modified by redistributing available material in an iterative process to produce an optimal design. The exact topology optimization method used in this thesis is the solid isotropic material with penalization (SIMP) scheme [1].



**Figure 1-2.** General sequence of a topology optimization, with iterations shown.

In this method, material is distributed by modifying the densities of elements in the mesh. To clarify, each element can exist as a fully solid element, a void, or somewhere in-between. This is known as an element's density. Note that this does not refer to a physical density, but rather the applied proportion of the element's Young's modulus. A solid element contributes its entire stiffness, while a void element essentially offers none. When a volume fraction is applied as a constraint, it forces the design to use a maximum percentage of the total initial volume of the design domain. Therefore, during optimization, the available material is redistributed throughout the domain, forming the final geometry, as seen in Figure 1-2. To have a design with mostly solid and void elements (known as black-and-white or 1-0), a penalty,  $p$ , is applied to those elements which have intermediate densities, hence the penalization portion of SIMP. The compliance is represented as  $c$ , element density as  $\rho$  and the number of elements as  $N$ . The elemental displacement vector and stiffness matrix are  $u_e$  and  $k_e$  respectively, with the global representations as  $U$  and

$K$ . The global force vector is  $F$ , the volume fraction is  $f$ , with  $V$  and  $V_0$  representing the material and design domain volume. The can be formulated as:

$$\begin{aligned}
 \text{Minimize Compliance: } c(\rho) &= U^T K U = \sum_{e=1}^N (\rho_e)^p u_e^T k_e u_e \\
 \text{Subject to: } \frac{V(\rho)}{V_0} &= f \\
 ; \quad K U &= F \\
 ; \quad 0 < \rho_{min} &\leq \rho \leq 1
 \end{aligned} \tag{1-1}$$

To solve this problem, an update scheme is used. Here, the optimality criteria method is used, with  $m$  as the move limit, and  $\eta$  ( $= 0.5$ ) as the numerical damping coefficient:

$$\rho_{e_{new}} = \left\{ \begin{array}{l} \text{if } \rho_e B_e^\eta \leq \max(\rho_{min}, \rho_e - m), \\ \quad \mathbf{max}(\rho_{min}, \rho_e - m) \\ \text{if } \max(\rho_{min}, \rho_e - m) < \rho_e B_e^\eta < \min(1, \rho_e + m), \\ \quad \rho_e B_e^\eta \\ \text{if } \min(1, \rho_e + m) \leq \rho_e B_e^\eta, \\ \quad \mathbf{min}(1, \rho_e + m) \end{array} \right\} \tag{1-2}$$

$$B_e = \lambda^{-1} \frac{\partial c}{\partial \rho_e} \tag{1-3}$$

$$\text{Sensitivity: } \frac{\partial c}{\partial \rho_e} = -p(\rho_e)^{p-1} u_e^T k_e u_e \tag{1-4}$$

The optimality condition determines  $B_e$ , with the term  $\lambda$  representing a Lagrangian multiplier that is found using the bi-section method. The sensitivity of the objective function based on the design variable  $\rho$  is also shown. This represents the change in compliance when the current element is removed. The sensitivity also includes the penalization term  $p$  to help converge the solution towards a 0-1 design. In general,

the higher an element's sensitivity, the denser the element will become in the next iteration. The opposite is also true for low sensitivity elements.

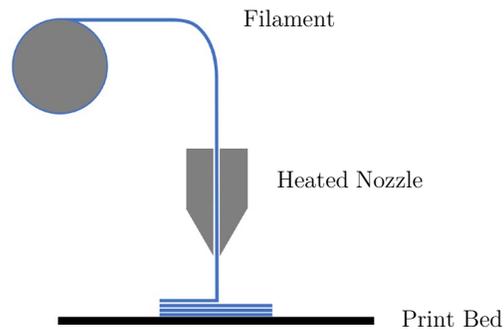


**Figure 1-3.** Checkerboard effect.

This process is repeated in an iterative manner, where an element's compliance is found, its corresponding sensitivity calculated using Equation 1-4, and the element's new density found using Equation 1-2. To be considered converged, the change in element densities between iterations is calculated, and if the difference is less than a set value (typically 1% or 0.1%), the iteration is terminated, and the solution presented. One issue that arises during a typical topology optimization run is the formation of discontinuities in element densities along edges. This leads to what is known as the checkerboard effect, seen in Figure 1-3, named after the tiled appearance of alternating solid and void elements. This phenomenon leads to issues when attempting to manufacture the design, or when deciding how to proceed with the optimized geometry. To correct for this, a simple filtering technique is applied during each iteration of the topology optimization. Here, element values are averaged by their neighbour's values, in a weighted scheme based on the distance to the next element. This effectively solves the checkerboard problem, and helps the problem remain mesh-independent, as in, results do not change significantly based on further refinement of the mesh.

### 1.1.3 Additive Manufacturing

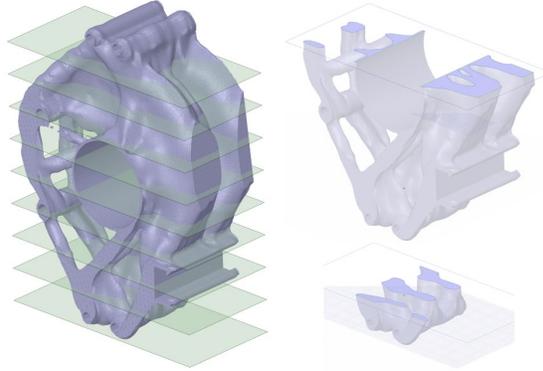
For this thesis, additive manufacturing will refer to layer-based AM, the most prevalent form of the technology. Specifically, it is relevant to AM processes where material is deposited in traced lines, known as toolpaths, for each layer of the component. The most common form of this is fused deposition modeling (FDM), also known as fused filament fabrication (FFF), where a thermoplastic filament is deposited through a heated extrusion process onto a bed, which is moved down relative to the extruder for every new layer (Figure 1-4). Another relevant process is selective laser sintering (SLS) and related powder bed technologies, where a laser sinters polymer particles together in traces.



**Figure 1-4.** FDM/FFF Process.

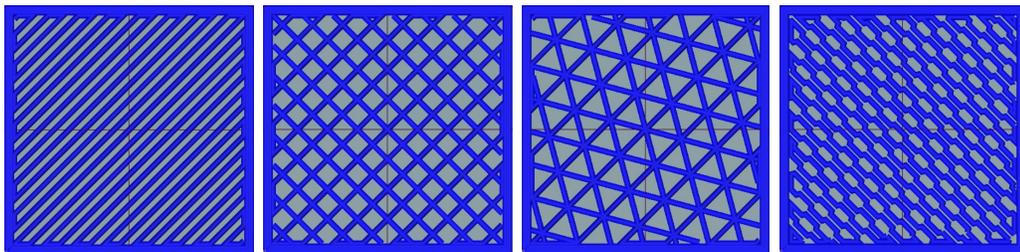
In order to generate toolpaths for the material deposition, the geometry to be printed must undergo a process known as slicing. The part geometry will be created in a computer aided design or engineering (CAD/CAE) software, and then it will generally need to be converted to what is known as a stereolithography file format (STL). From here, the STL will be sliced into cross-sections based on the chosen direction, and layer thickness for the specific AM process and surface quality desired (Seen in Figure 1-5). These slices will contain the boundary curves of the geometry

at the specific slicing height. Then, within each slice, a set of toolpaths will be created and converted to g-code for a 3D printer to interpret.



**Figure 1-5.** Depiction of slicing an STL.

For the AM methods introduced, the most important toolpath modification is the concept of infill. Here, to save on printing time and material, the component is often not printed fully solid. Instead, several exterior walls, or perimeters, are deposited every layer with the interior being filled with a pattern that reduces the material used, while keeping the required structural performance. Common infill patterns include rectilinear (zig-zag), grid, triangle, and honeycomb patterns as seen in Figure 1-6. It is important to note that due to the reduced effective surface area of the cross section, the perimeter will become significantly more influential on the total time taken to print the layer.



**Figure 1-6.** Various infill patterns from a popular open source slicer (Slic3r).

# Chapter 2

## Literature Review

### 2.1 Overview of Topology Optimization Methods

Topology optimization was first introduced in 1988 by Bendsøe and Kikuchi [2], building on earlier works of Cheng and Olhoff [3], and others working in homogenization theory. This early result introduced a method in which material is distributed throughout a domain in a computational scheme. Since then, much research has been conducted exploring the topic, and many variations and branches of the material distribution method have been developed. One of the most popular approaches to topology optimization, the density-based approach, was set by the key research of Bendsøe [4], Zhou and Rozvany [5], [6], and Sigmund [1], [7] from the late 1980's through to the early 2000's. This method, as described above, discretizes a domain into elements, then modifies the density of these elements to form the optimal structure. Notable versions of this method include the solid isotropic material

with penalization (SIMP) scheme used in this thesis, as introduced by Bendsøe, Sigmund, Zhou, Rozvany, and Mlejnek [4], [6], [8], and variations of the evolutionary structural optimization (ESO) method of Xie and Steven [9]. The level set method is a distinctly different approach and was introduced in 2002 by Allaire and Wang [10]–[12]. It differs from the density-based approach due to the use of shape derivatives instead of discrete elements in the optimization process. A level set function is described in an initial guess that will include a certain number of features, and this is then modified along the existing boundaries until it forms the final structure. It perhaps is closer to that of a true shape optimization, but with a much stronger ability to modify existing geometry. Some other notable approaches include the topological derivative method of Sokołowski and Żochowski from 1999 [13], and Bourdin and Chambolle’s phase field method from 2003 [14].

## **2.2 Manufacturing Constraints in Topology**

### **Optimization**

Before the prominence of additive manufacturing and AM constraints for topology optimization, conventional manufacturing constraints were explored in research. Topology optimized parts often break fundamental manufacturing rules, such as creating voids inside a solid structure or requiring impossible cutting operations. This is because a normal topology optimization does not account for the manufacturing process that will be used, instead creating the most optimal shape regardless of geometry. Therefore, a post-processing step that modifies the optimized geometry to

correct these issues must be performed. However, this may severely compromise the designs optimality without significant investment to re-optimize the part. Here, the main contributions of conventional manufacturing constraints that are also relevant to AM are explored.

### **Feature Size**

One manufacturing constraint that was developed early on is the minimum member or feature size. Also known as length control, this constraint helps ensure that any feature in the topology optimized structure is larger than a certain size. If a member is too small, machining can become difficult due to issues with cutter sizes and material characteristics. For AM technologies this can be thought of as the smallest area on which material can be deposited. The method for controlling this was developed from the same method used for avoiding checkerboard patterns and ensuring a black-and-white design, as introduced by Sigmund, Petersson, Bourdin, Bruns, and Tortorelli in the late 1990's [15]–[18]. The technique for explicitly constraining feature size uses the Heaviside projection method of Guest et al. [19] which modifies the element densities based on a projection of the nodal values with a minimum length applied. Further work by Guest was done to ensure that not only do solid members follow this rule, but also the interior voids, as both may be important for manufacturing constraints [20]. More recently, image based computational methods were used by Zhang et al. [21] to explicitly control minimum and maximum lengths by extracting a structural skeleton from the model and applying the length scale to that. Geometric constraints have also been used directly

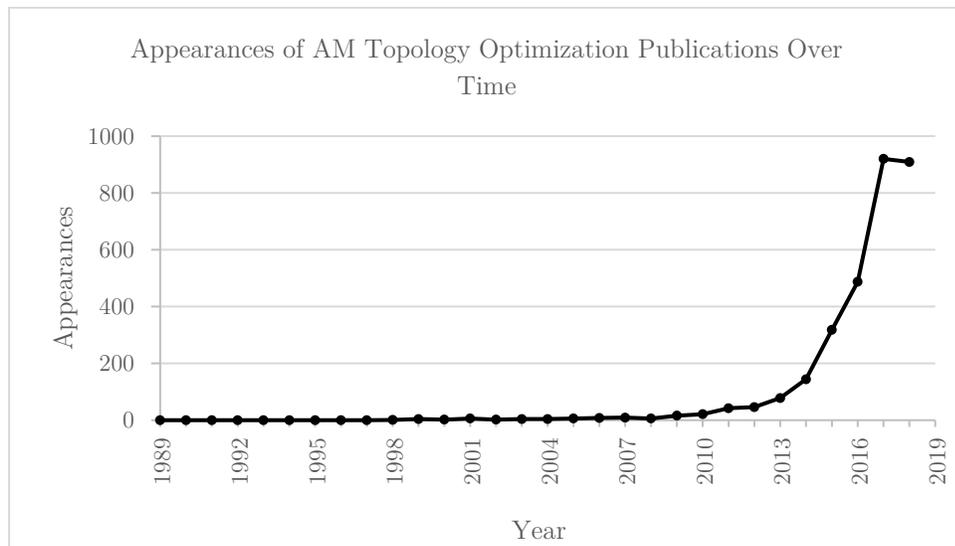
in the optimization scheme by Zhou et al. [22] to limit the maximum and minimum lengths.

### **Internal Cavities**

Another set of features that are considered during topology optimization are internal cavities or voids. For machining or casting purposes these cavities are impossible to manufacture; and for powder or liquid based AM processes they would trap material used in the manufacturing process. Early research works in the 2000's by Schramm and Zhou et al. [23]–[25] considered casting and extrusion manufacturing constraints, successfully limiting void formation by only allowing elements to completely change density in the direction of casting. Though, this is more of a side effect of the constraint, since the design will have a completely consistent cross-section in the drawing direction. A more specific constraint for void formation during topology optimization was developed by Liu et al. and Li et al. [26], [27] in the mid 2010's. This method applied an imaginary heat source to void elements. Here, internal cavities will have naturally elevated temperatures compared to exposed voids, as heat transfer will be limited by the nature of the enclosed cavity. This allowed only internal voids to be penalized throughout the optimization, leading to their elimination.

## 2.3 Topology Optimization for Additive Manufacturing

Recently, there has been an explosion in research on topology optimization for additive manufacturing (Figure 2-1). Since a key patent for fused deposition modeling expired in 2009, the process has become exponentially more accessible to companies, hobbyists, and researchers. Combined with an increase in computational power and prevalence of topology optimization, the two have become heavily intertwined. Here, the current research efforts in topology optimization for additive manufacturing will be detailed.



**Figure 2-1.** AM and topology optimization research papers over time (based on appearances in Google Scholar).

### Overhanging Features

For many AM processes, one of the limitations in manufacturability is the build angle of the component. Once a certain angle is reached (generally 45 degrees relative to the build platform) or the feature is completely overhanging, the material being

deposited no longer has anything to support it, leading to a likely print failure. Currently, to mitigate printing issues with these overhanging features, sacrificial support structures are generated to enable geometry that does not comply with the angle limit. These supports add time and material to the printing process. They also negatively affect the surface quality of the part since they inherently need to be connected to the final part, then removed in a post-processing step [28].

The first notable approach to reduce the amount of support material used was introduced by Allen and Dutta in 1994 [29]. In a simple scheme, multiple orientations were searched to find the one with the least amount of support material. A modern version of this work was developed by Morgan et al. in 2016 [30], and a more complete optimization scheme accounting for surface quality in difficult and visually significant areas was done by Zhang et al. in 2015 [31]. A logical step from this was to try to reduce the amount of support material being generated for the same geometry and orientation. Early works in 2009 by Huang et al. [32] slimmed down the standard rectangular supports using sloped walls. In the mid 2010's Vanek et al. and Gan et al. [33], [34] used a more complex scheme to produce organic tree structures that significantly cut down the amount of material used. More structurally robust techniques include the scaffolding method of Dumas et al. [35], and the less printer friendly use of cellular structures by Strano et al. [36].

The more interesting approach, at least with relevance for this thesis, is to combine the goals of overhang free design directly into the topology optimization process. This idea was first proposed and detailed by Brackett et al. [37] in 2011. The first attempt to implement this was in the form of a post-processing step by

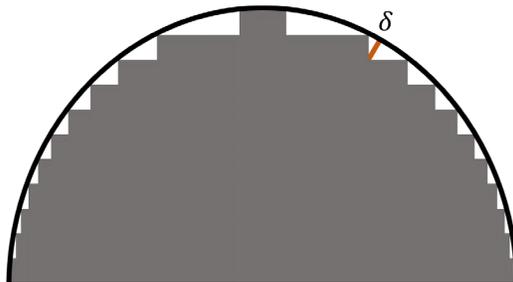
Leary et al. in 2014 [38]. Here, a completed topology optimized geometry was analyzed, and modifications were automatically made to remove any overhanging geometry. Of course, the issue here is the same as manually fixing the part; the optimality will almost certainly be compromised in the process. The first to successfully achieve an overhang free design through topology optimization were Gaynor and Guest in 2016 [39]. Based on an earlier attempt [40], overhanging elements were penalized in a similar way as the length scale method, in that it uses a support region Heaviside projection to ensure the final geometry was without overhanging features at any arbitrary angle in 2D. The system was layer-wise, where the constraint was applied directionally, going up through the layers from the build platform. Later, Langelaar [41], [42] detailed a similar density layerwise filtering method, also extending it to 3D. Note that while this method produced similar results to Gaynor and Guest's, it was not mesh independent like theirs. Qian [43] also demonstrated a filter that used a density gradient with a Heaviside projection to limit the overhanging features in 2D and 3D.

In 2016 Mirzendehdel and Suresh [44] applied a constraint to not completely remove overhanging structures, but rather minimize them through penalization of possible support structures. A similar approach to only partially remove overhanging material was detailed in [45], [46]. Another approach used two sequential structural optimizations to first solve a simple skeleton truss-based model to comply with the overhang constraint, then solve a topology optimization based on that to complete the material distribution. This approach did not fully remove overhanging constraint, and was recently detailed by Mass and Amir [47], [48]. A simultaneous optimization

orientation and overhang constraint to reduce the required support material was also demonstrated in a limited fashion by Driessen in 2016 [49], and later extended by Langelaar in 2018 [50].

### Slicing

While not a topology optimization constraint, it is useful to examine research that has been done in slicing, as the following topics will utilize such methods. Since layered manufacturing (LM) processes by definition will have some finite layer height, the final produced geometry will only approximate the initial model. This is known as the staircase effect, with the error quantified by the cusp height (seen in Figure 2-2):



**Figure 2-2.** Staircase effect and cusp height in orange ( $\delta$ ). Black is the boundary of the original model, grey is the deposited material.

This concept was pioneered by Dolenc and Mäkelä [51], who attempted to minimize cusp height by adjusting slice geometry to compensate. Generally, these 3D printing process used the same layer thickness throughout the model. However, it is clear that as the relative angle of the geometry increases, the cusp height will also increase. To alleviate this, the layer thickness can be made to be much smaller, coming at the cost of increased printing time. Therefore, methods to adaptively select slicing

heights were developed initially by Sabourin et al. [52] and Tyberg and Bohn [53], and later expanded by Cormier et al. [54]. These methods reduced cusp error by selecting smaller layer heights when required, otherwise using larger heights to improve the printing time.

It is standard for slicers to use tessellated CAD geometry in the form of STL files, although this carries some issues. The conversion to STL can lead to inaccurate parts due to the creation of non-manifold geometry, degenerate faces, gaps, overlaps, and other errors in the model. These will also generally result in an increased computation time for slicing. The tessellation process itself will also add tolerance error due to its approximation of the original CAD geometry using triangles. The number of triangles used determines the accuracy of the model, and as the number of triangles increases, the computational time to produce and slice the STL increases proportionally. Some early research by Jamieson and Hacker [55] showed a viable method for slicing CAD models directly to reduce these issues, and this was also demonstrated by Zhao and Laperrière [56]. More recent work by Sikder et al. [57], Barari et al. [58], and Gohari et al. [59] involves slicing the CAD model based on extracted parametric surfaces to improve tolerancing, and further optimizing the slices to reduce error. A method to form slices directly during a topology was first shown by Liu et al. in 2018 [60], however this used the level set method, where the geometry is explicitly defined. A more relevant method was recently shown by Bender and Barari [61], where an FEA mesh was directly sliced.

**Anisotropy**

Due to the layer-based nature of AM processes, parts will often be weakest in the direction of printing, as well as have further anisotropy based on the toolpaths taken during deposition [62]. In 2013, Umetani and Schmidt [63] looked into the critical stress points created by the anisotropic properties of 3D printed objects, and optimized the build orientation to reduce these. Further investigation was done by Ulu et al. to optimize for the stress tensor based on build orientation. A study detailing the anisotropic properties of a topology optimized part was conducted by Liu [64]. Mirzendehtdel et al. [65] used the Tsai-Wu stress criterion instead of the typical von Mises to account for anisotropic properties, which lead to stronger, more robust topology optimized parts. The work of Smith, Høglund [66] and Jiang [67] to develop a method to optimize the orientation of polymer fibres deposited during the FDM process alongside a topology optimization is also relevant. The first work to incorporate toolpath planning in topology optimization was described by Liu and Yu in 2017 [68]. This set a fixed method of toolpath generation and incorporated the resulting anisotropy into the topology optimization process. This was extended to include hybrid selection of toolpaths [60], and then to include the use of both AM and subtractive methods [69]. This work was also extended to other toolpath methods by Dapogny et al. in 2019 [70].

**Infill Optimization**

As mentioned in the background, certain AM processes can vary the amount and geometry of solid infill used. This can help minimize a component's weight while

keeping acceptable structural performance, and also reduce the time to print. The process of slicing and then generating different interior and exterior toolpaths can be traced back to Sabourin et al. in 1997 [71]. For topology optimization, building on work from 1998 by Sigmund et al. [72], the first significant effort to concurrently optimize a structural component using microstructure “infill” was completed by Sigmund in 2000 [73]. A similar result for producing optimal microstructures during topology optimization was obtained by using a hierarchical procedure by Rodrigues et al. [74]. This work was extended to physical tissue regeneration using topology optimization to produce AM built scaffolding with suitable stiffness and permeability in a landmark paper by Hollister in 2005 [75]. Later, Coelho et al. [76] extended the general microstructure work, leading to the inclusion of biomaterial laminated composites to the process [77]. The use of skin frame structures combined with internal scaffolding was also demonstrated for structural purposes by Zhang et al in 2015 [78]. Rhombic self-supporting infill structures were then demonstrated by Wu et al. [79], and later expanded to porous bone-like infill [80]. A variable internal lattice structure that closely resembles general purpose infill was also demonstrated in a topology optimization process by Liu et al. [81] and Wu et al. [82] in 2017. Shell structures with internal infill were also explored notably in 3D by Clausen et al. [83].

### **Cost & Time Constraints**

As previously stated, one of the major limiting factors for mass production using additive manufacturing is the cost. General purpose printers using layer-based methods will inherently take longer than custom tooling in conventional methods

such as injection molding and casting. Other than reducing cost via minimization of support material, early work by Huang et al. in 2017 [84] showed that cost of metal 3D printing could be added in topology optimization by including parameters for material cost, manufacturing cost, energy cost, and scrap & idle time cost. While not optimizing directly for build time, due to improvements in optimal laser speed and power settings, build time was reduced by 51% and overall cost decreased by 7% in a single idealized 2D example. It should be noted that material use increased by 31% compared to an unconstrained topology optimization, therefore this approach would not translate well to FDM or polymer powder sintering based processes since they do not have the same melt-pool size and pass functions metal printing does. They also do not comment on the effect on compliance. This work was further extended by Liu et al. in 2019 [85], now using material cost, argon cost, and operational cost as constraints in the topology optimization. The effect was to reduce part height and from this they were able to achieve a cost reduction of 15% in a 2D example, and a 4.6% reduction in a 3D example.

One step to help mitigate lengthy build times expands on the concept of infill and topology optimization. Since the component shape is generated dynamically, it stands that one can constrain the outer perimeter, or shell, to minimize the printing time needed. During the completion of the author's work on this topic in 2018-2019 [86]; Sabiston, Ryan, and Kim [87], [88] introduced this idea via the use of spatial gradient information based on the densities in the discretized mesh. The work used this information to identify elements that lay on the perimeter during each iteration, then applied a weighted penalty to these elements in order to decrease the perimeter.

This was completed in 2D, and the resulting reduction in print time over two examples was 10.7% and 12.1% compared to a typical topology optimization. In this work, the addition of support structure minimization was also completed using traced rays to identify overhanging elements, and again, a weight penalty was applied to these. It should be noted that no examples of both constraints being added at the same time were shown in this work.

## 2.4 Gaps in the Literature & Thesis Objectives

From the literature survey, it is clear that support structure minimization during topology optimization is a well-established constraint. Here, it is the author's opinion that effort should be continued on establishing a balance between compliance (or any other topology optimization goal) and reduction in support structures, rather than total support elimination. On the subject of anisotropy, if topology optimization for AM is to become robust enough for commercial application, both print direction and generated toolpaths need to be accounted for. Microstructure infill also has the ability to produce extremely light designs with substantial structural performance, however, issues in printing time and reliability of such small features seem to be holding this back from a commercial perspective.

Looking at the research, the key factor encompassing AM constraints for topology optimization is fabrication time. While minimization of support structures and the use of optimized infill certainly contribute to this, a clear design for AM approach would be to explicitly consider print time during the optimization. The work completed so far on this topic seems promising, but it has not been extended

to 3D, and thus does not consider the layered effect of AM processes. Also, in the literature it is noted that when topology optimization code is offered as additional material, it is almost exclusively written in MATLAB or another high-level scripting language. While these programming languages are commonplace in engineering and the sciences due to their ease of use, the popular implementations of topology optimization offer very limited mesh and boundary condition setups, with simple FEA solvers. Therefore, the objectives of this thesis can be summarized here:

- Implement a slicing mechanism in this platform to directly retrieve the layered contours, or perimeters, seen in additive manufacturing. In order to avoid conversion from STL to FEA and back every iteration, a direct slicer will be used. It must work for 3D problems and identify and categorize elements that lie on the contours of the component.
- Use this information to develop a new additive manufacturing topology optimization constraint on the perimeter of the geometry, with the goal of minimizing print time in a layer-based AM process.
- Build a platform to perform robust customized topology optimization while allowing for simple integration of existing and future research code.

# Chapter 3

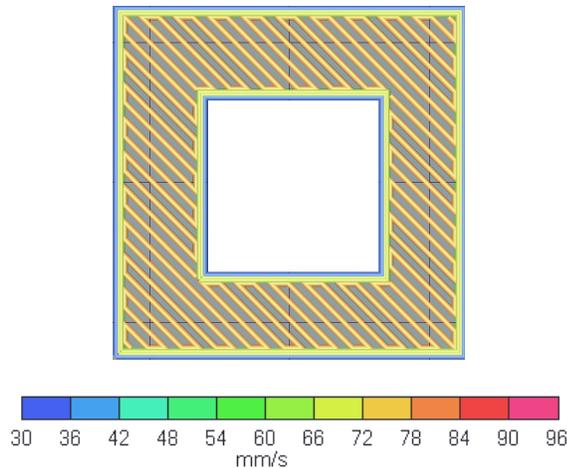
## Methodology

This chapter will detail the proposed additive manufacturing topology optimization perimeter constraint. In it, the theory behind reduction of print time will be established, and the “mass concentration” approach will be described and formulated in a new topology optimization problem. The method of slicing and categorizing the resulting boundary contours will also be detailed.

### 3.1 Reduction of the Perimeter

As mentioned in the previous chapter, generally, a layer’s print time is not truly represented by its unmodified surface area. If infill is used, the actual amount of material to be deposited in the section will be reduced. In addition to this, the exterior and interior boundaries of the layer will be traced as an outline, generally in 2-3 passes. In most, if not all slicing and toolpath generating software, the printing speed of these outlines, or perimeters, is also reduced compared to that of the interior

infill. Generally, the outer most walls have a speed reduction of around 50%, with subsequent outline passes increasing in speed. This can be seen in Figure 3-1. Thus, a slice's outline, its perimeter, will determine a larger proportion of the time it takes to print compared to an unmodified layer. Ideally, both perimeter and surface area would be optimized in some sort of weighted scheme, but to the author's knowledge, there is no way to minimize internal surface area directly. Therefore, minimizing the perimeter will be used as a means to reduce the printing time.



**Figure 3-1.** Print speed visualization of perimeters and infill (Generated by the open-source *Slic3r*).

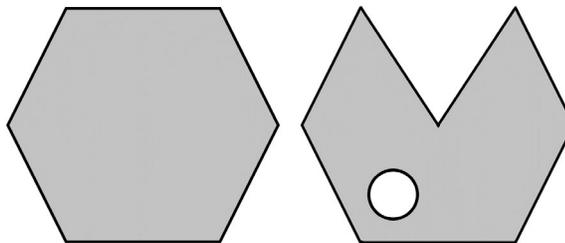
### 3.1.1 Ideal Perimeter

It is useful to first define what an ideal perimeter geometry would look like in terms of the cross-sections to be examined. That is, for a given area, what shape would produce the minimum perimeter. Another popular way to frame this question is to determine the shape with the largest area for a given perimeter. Such a question is perhaps the oldest calculus of variations problem, first described by Dido's problem in the *Aeneid* by Vergil in 19 BC [89]:

“The Kingdom you see is Carthage, the Tyrians, the town of Agenor;  
 But the country around is Libya, no folk to meet in war.  
 Dido, who left the city of Tyre to escape her brother,  
 Rules here - a long and labyrinthine tale of wrong  
 Is hers, but I will touch on its salient points in order...  
 Dido, in great disquiet, organised her friends for escape.  
 They met together, all those who harshly hated the tyrant  
 Or keenly feared him: they seized some ships which chanced to be ready...  
 They came to this spot, where to-day you can behold the mighty  
 Battlements and the rising citadel of New Carthage,  
 And purchased a site, which was named 'Bull's Hide' after the bargain  
 By which they should get as much land as they could enclose with a bull's hide.”

This is known as an isoperimetric problem, with the first geometric proof coming from Steiner in 1841 [90]. Later, more complete analytic proofs were formed [91]. Based on these, it can be said that for any arbitrary closed shaped, a circle contains the most area for a given perimeter; or a circle has the smallest perimeter for a given area. While not intended to be a rigorous proof, the basics of this theorem will be covered.

To start, a concept important to this problem is geometric convexity. A region can be defined as convex if any two points on the shape form a line that fully exists inside the shape. By this definition, a convex shape does not contain any voids or holes. Examples of both types can be seen in Figure 3-2:



**Figure 3-2.** Convex shape (left) and a non-convex shape (right).

It can be said that for any closed region, adding a cavity will increase the perimeter for a given area. Suppose region  $S_1$  exists with area  $A_1$  and perimeter  $P_1$ . Then, a cavity of area  $A_c$  and perimeter  $P_c$  is added to form region  $S_2$  with the total area and perimeter as:

$$A_2 = A_1 - A_c \tag{3-1}$$

$$P_2 = P_1 + P_c \tag{3-2}$$

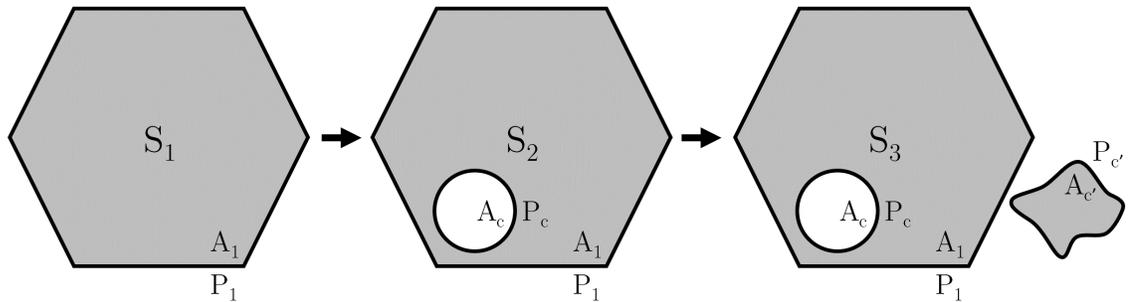
To compensate for the lost area, an addition region is added with area  $A_{c'} = A_c$ , and arbitrary perimeter  $P_{c'}$ , forming region  $S_3$  (seen in Figure 3-3):

$$A_3 = A_2 + A_{c'} = A_1 - A_c + A_{c'} = A_1 - A_c + A_c = A_1 \tag{3-3}$$

$$P_3 = P_2 + P_{c'} = P_1 + (P_c + P_{c'}) \tag{3-4}$$

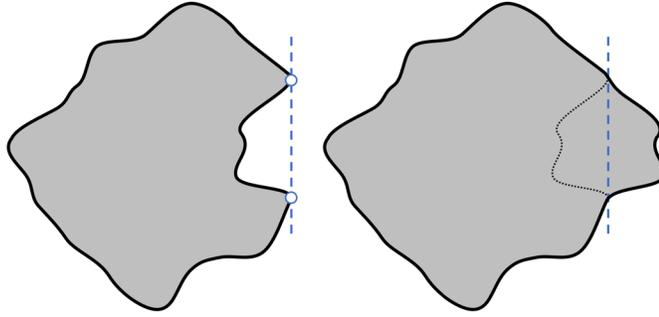
And thus:

$$P_3 > P_1 \tag{3-5}$$



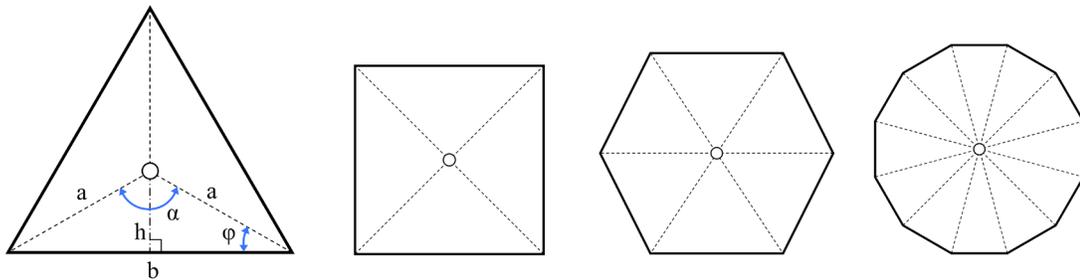
**Figure 3-3.** Process showing increase in perimeter when adding a cavity

It can also be said that if region  $S$  is not convex while containing no cavities, two points forming a line outside of  $S$  will always exist. Taking the mirror of the outline contained by these two points, it can be seen that the perimeter has not changed, but the area of  $S$  has increased (Figure 3-4). Thus, a region must be non-convex in order to satisfy the theorem.



**Figure 3-4.** Non-convex mirroring process.

From here it is more intuitive to examine polygons with equal length edges. Hence, a proof will be formed as; for any regular convex polygon with  $n$  number of edges, the polygon with the minimum perimeter for a given area will contain the most edges. This will most closely approximate a circle. Note that each polygon in this setup will have a centre point that is exactly the same distance to any vertex on the polygon as seen in Figure 3-5:



**Figure 3-5.** Regular polygons with 3, 4, 6 and 12 sides with approximately the same area.

The distance to the vertex from the centre point can then be defined as,  $a$ . If lines are drawn from the centre of the polygon to each of its vertices, isosceles triangles are formed from these lines and the edges of the polygon. The height of these triangles is,  $h$ , and with the edge length as,  $b$ , the perimeter of the polygon can be defined as:

$$P = n * b \quad (3-6)$$

The area of one isosceles triangle can then be found using:

$$A_{isosceles} = \frac{b h}{2} \quad (3-7)$$

The height,  $h$ , can be related to the apex angle of the triangle,  $\alpha$ , by using cosine to find the adjacent length. The angle  $\alpha$  is found by taking the full angle  $2\pi$  and dividing that by the number of edges. To use the cosine relation to find  $h$ , the isosceles triangle will be split in two, therefore the resulting angle will be half of  $\alpha$ , and will be known as  $\theta$ .

$$\theta = \frac{\alpha}{2} = \frac{2\pi}{2n} = \frac{\pi}{n} \quad (3-8)$$

$$\cos(\theta) = \frac{h}{a} \quad (3-9)$$

$$h = a \cos(\theta) \quad (3-10)$$

To get  $a$  in terms of the base length, the same cosine procedure is done for angle  $\varphi$ .

$$\varphi = \pi - \frac{\pi}{2} - \theta = \frac{\pi}{2} - \frac{\pi}{n} \quad (3-11)$$

$$\cos(\varphi) = \frac{b/2}{a} \quad (3-12)$$

$$a = \frac{b}{2 \cos(\varphi)} \quad (3-13)$$

This can then be substituted into Equation 3-7:

$$A_{isosceles} = \frac{b a \cos(\theta)}{2} = \frac{b \frac{b}{2 \cos(\varphi)} \cos(\theta)}{2} \quad (3-14)$$

To get the total area of the entire polygon, Equation 3-14 is multiplied by the number of edges (corresponding to the number of isosceles triangles). The base length  $b$  and the angles are also put in terms of  $n$ .

$$A_{polygon} = \frac{\frac{P}{n} \frac{P/n}{2 \cos(\varphi)} \cos(\theta)}{2} n = \frac{\frac{P}{n} \frac{P/n}{2 \cos(\frac{\pi}{2} - \frac{\pi}{n})} \cos(\frac{\pi}{n})}{2} n \quad (3-15)$$

And simplified:

$$A_{polygon} = \frac{P^2 \cot(\frac{\pi}{n})}{4 n} \quad (3-16)$$

With Equation 3-16, it can be said that the area of the polygon will be maximized as  $n \rightarrow \infty$ , which would of course, approximate a circle. Therefore, the shape which maximizes area for a given perimeter would be a circle. Conversely, this holds true for the shape with the minimum perimeter for a given area.

### 3.1.2 Perimeter Normalization

In a SIMP-based topology optimization, it is important to remember that each iteration is constrained by a set volume fraction as seen in Equation 1-1. This means that the material distribution, or the density of each element, must only sum up to the prescribed value. Initially, elements will have a variety of low densities, allowing the material to explore all regions of the design domain. However, as the optimization converges, element densities will go towards a black-and-white design, with little in-between. In a converged 2D slice, this means the volume fraction will be equivalent to the surface area of the design. Therefore, based on the exact distribution of material in the design, the main factor differentiating different designs will be their perimeters.

To quantify the performance of different cross-sections perimeters in relation to each other, the perimeter will be normalized based on the ideal perimeter for the

given surface area. This will allow the final perimeter to be dimensionless and make it easier to compare different cases with each other to judge their relative performance. To do this, the perimeter of the current slice,  $P_i$ , will be divided by the perimeter of a circle for the same area,  $P_c$ . The formulation of  $P_c$  is based on the area and perimeter formulas for a circle, with  $A_c$  as a circles area:

$$A_c = \pi r^2,$$

$$r = \sqrt{\frac{A_c}{\pi}} \quad (3-17)$$

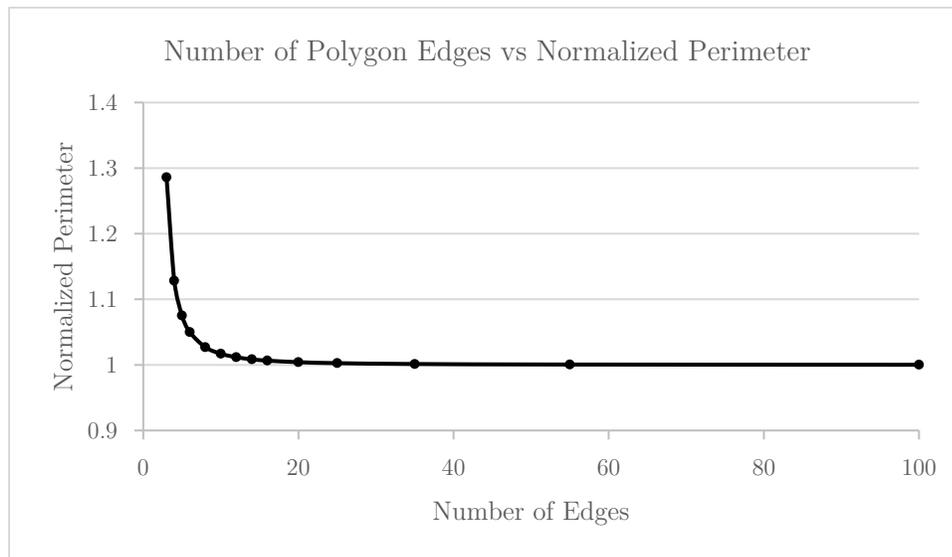
$$P_c = 2\pi r$$

$$P_c = 2\pi \sqrt{\frac{A_c}{\pi}} = 2\sqrt{A_c\pi} \quad (3-18)$$

The normalized perimeter will then be:

$$P_N = \frac{P_i}{P_c} \quad (3-19)$$

Using Equations 3-16 and 3-19, it can be seen that as the number of edges increases, the normalized perimeter goes to 1 (Figure 3-6). Which would equate to  $P_i = P_c$ .

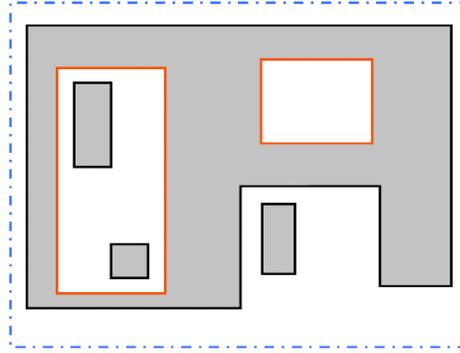


**Figure 3-6.** Plot showing how the number of edges of a polygon

### 3.1.3 Mass Concentration Approach

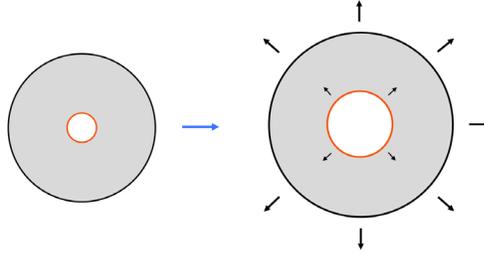
Given the knowledge that a circle is the most ideal shape for minimizing perimeter, the basis for a method to minimize the perimeter in a topology optimization can be established. Beyond the theorem proofs, nature provides examples of this with hexagonal shapes beehive cells; and in three dimensions, planets and celestial objects. It is apparent that the concentration of a circles area on its centre is responsible for its ideal properties. Due to this, could it be stated that the ideal perimeter-based topology optimization simply creates the most circular object possible for every design domain? Of course, just as an egg is not spherical, other factors must be accounted for as well. The optimization must contend with stiffening the geometry based on applied boundary conditions, and simple, circular shapes would not be conducive to the vast majority of engineering problems. Thus, the solution will have to be more subtle. In order to reduce print time by minimizing a slices perimeter, the proposed optimization constraint will attempt to concentrate area (or in an engineering problem, mass) of a slice using two principles; outer perimeter reduction and inner perimeter collapsing. This method will be known as the mass concentration approach.

First, the difference between an outer and inner perimeter must be defined. An outer perimeter is the contour that exists on the outer boundary of a shape, whether that be the main geometries shape, or, a shape within a void in the geometry. An inner perimeter then, is the contour that exists on any inner boundaries that exist in the geometry. The two perimeters can be seen in Figure 3-7:

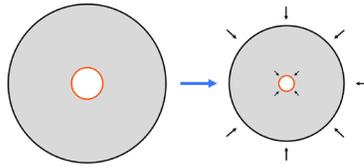


**Figure 3-7.** Design domain in blue. Solid area is grey and white is void. Outer perimeters are shown in black, and inner perimeters in red.

It should be noted that while inner perimeters may not exist, an outer perimeter will always exist. With this, the two types of perimeters can be differentiated. These are distinguished due to the different effects caused by modifying the surface area around them. Two modification can be done at each perimeter; material can be added or removed along both sides of the contour. If material is added at the outer boundary contour, this will cause the outer perimeter to increase (black contour in Figure 3-8). However, if material is added to the inner boundary contours, the inner perimeter will decrease (orange contour in Figure 3-9). Conversely, if material is removed from the outer contour the perimeter will decrease, and if material is removed from the inner contour it will increase. Therefore, it is thought that the optimal way to decrease the perimeter of a shape will be to remove material around the outer boundaries and add material around the inner boundaries. The resulting effect will be the concentration of surface area, or mass, due the collapsing of inner voids, and the shrinking of exterior surfaces.



**Figure 3-8.** Graphic of expanding perimeters.



**Figure 3-9.** Graphic of shrinking perimeters.

## 3.2 Identification of Boundary/Perimeter Contours

Now that a method to decrease an arbitrary shape's perimeter has been established, its contour boundaries must be identified and classified. Here, a slicer will be used to identify the contour edges, with a ray-tracing approach for classification of inner and outer boundaries. Specifically, a current state-of-the-art finite element slicing algorithm, as discussed in the literature review, will be modified for use in this thesis.

### 3.2.1 Slicing

As mentioned previously, since topology optimization occurs in an FEA simulation process, it is beneficial to use the finite element mesh directly for slicing. Due to the iterative nature of the optimization, the typical STL slicer would require many conversions, increasing the risk of error accordingly. Therefore, a current finite element slicer from Bender will be adapted [61]. In this slicing process, first the

elements are culled based on their position in the mesh. Here, elements that do not intersect with the slice plane are eliminated. Next, elements that do not meet the required density to be considered solid are removed. This can be adjusted but is typically set at around a density of 0.9, on the 0-1 scale. From the remaining elements, the intersection of edges with the plane is determined. For the standard cuboid or tetrahedral element, this will be 3-4 intersections. These edges are then connected together to form the initial set of line segments in the plane. From here, the outer boundary contours are identified by determining which edges appear on the list of edges on the plane once, in other words the non-redundant edges. The slicing process can be seen in Figure 3-10. The solid element mesh with the slicing plane in pink is shown on the left, with the elements lying on the plane shown to its right. The first list of edge contours is shown at the top right, with the final slice contours shown below.

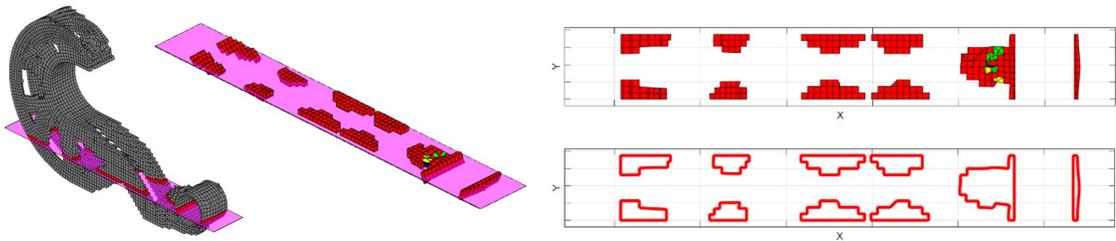
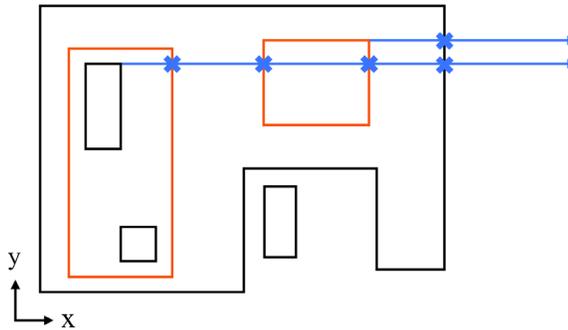


Figure 3-10. Slicing procedure [61].

### 3.2.2 Ray-tracing Approach

With the contours on the perimeter established, the next step is to identify which closed contours belong to the outside of the boundary and which belong to the inside. This is accomplished via a simple ray-tracing approach. By taking the right-most

edge of a single closed contour and drawing a line to positive infinity (i.e., a ray) along the x-axis, the number of intersections with other contours can be counted. If the number is even, this means that the initial contour is on the exterior. Conversely, if the total intersections are odd, this means the contour is on an interior boundary. This ray shooting process is then repeated for every contour. The process can be seen in Figure 3-11 below.



**Figure 3-11.** Ray-tracing for inner/outer contour detection. Outer contours in are black, with inner contours in orange. Rays are in blue, with intersections marked as blue crosses.

### 3.3 Formation of the New Topology Optimization

#### Constraint

Now that the approach to reducing a shapes perimeter is clear, and the perimeter contours are determined, the new additive manufacturing constraint must be included. Here, the SIMP method will be used due to its wide-spread adoption and popularity, as well as the relative ease of programming it. Since this method only includes compliance minimization with a constraint on volume fraction, as presented

in section 1.1.2, the problem must be reformulated. This section will detail the new terms added.

### 3.3.1 Perimeter Constraint

In a topology optimization, there are a number of ways to implement this perimeter constraint. How and where the constraint will be applied is the main question. Based on the proposed perimeter reduction scheme, in the author’s opinion it does not seem logical to explicitly force a perimeter reduction. Rather, the topology optimization should be guided to produce the desired result. As discussed previously, the goal will be to remove material around the outer boundary contours, and then add material along the inner contours. Thus, the aggressiveness of material removal and addition along boundary contours must be established.

In the most extreme way, a hard “kill” and “birth” method could be used to force the optimization to act on the perimeter constraint. This could lead to unstable convergence, or force the design into a local optimum, never allowing the design domain to be suitably explored. It is also predicated that constraint will have different effects based on the design domain and boundary condition setup. A better way, then, is to allow an adjustable weighted parameter to control the likelihood that an element along the perimeter will be removed or created. More specifically, if its density will be increased or decreased. Based on a typical SIMP formulation, element densities are modified based on their sensitivity in an update scheme. As discussed in the technical background, these sensitivities are based on the results of an FEA, with a penalization factor applied to force the elements toward a black-and-

white design. Therefore, it stands to reason that the element sensitivities will be modified before they are placed in the update scheme, allowing the effect of the constraint to be applied to element densities.

### 3.3.2 New Topology Optimization Formulation

To perform this sensitivity modification, a new total sensitivity,  $S_T$ , will be created. A new additive manufacturing (AM) penalty factor for the weighted application of the constraint,  $p_{AM}$ , is introduced in the sensitivity formulation.

$$\text{New Sensitivity: } S_T = \begin{cases} \frac{\partial c}{\partial \rho_e} + p_{AM} \frac{\partial c}{\partial \rho_e}, \\ \text{if the } \rho_e \text{ element appears on inner contour} \\ \\ \frac{\partial c}{\partial \rho_e} - p_{AM} \frac{\partial c}{\partial \rho_e}, \\ \text{if the } \rho_e \text{ element appears on outer contour} \end{cases} \quad (3-20)$$

$$\text{Subject to: } S_T > 0$$

The new sensitivity will replace the old sensitivity of Equation 1-4 in the update scheme of Equation 1-2. This method will be referred to as the AM topology optimization perimeter constraint, or just perimeter constraint, with  $p_{AM}$  being known as the AM perimeter penalty.

# Chapter 4

## Implementation

Numerical implementation of this system will be detailed, starting with the custom implementation of SIMP topology optimization in commercial FEA code. The slicing function will then be described, along with the determination of interior and exterior contours. Finally, the application of the AM topology optimization perimeter constraint will be discussed.

### 4.1 Implementation using Scripting & Commercial CAE Software

As mentioned, the SIMP topology optimization scheme will be used as a base for implementation of the new perimeter constraint. In most research, this is written in MATLAB as a stand-alone piece of code. The design domain, the application of boundary conditions, the meshing, the FEA, and the topology optimization are all contained within this. However, this leads to certain issues. While the MATLAB

environment is quite favourable for engineering researchers due to its high-level coding approach and thorough documentation, it also means that its usefulness in engineering applications is weak.

### **4.1.1 Justification for Utilizing Commercial FEA Code**

In the real world, almost all structural work is done in a CAD/CAE program suite, where a powerful graphical interface is used to setup complex problems, with robust meshing abilities, and a capable FEA solver. Here, multi-physics can also be used depending on the package, leading to even more complex and useful simulations. In contrast, with the research side of available code, problems are cumbersome to set-up, with limited graphical interfaces and boundary conditions often having to be hard-coded in. The meshing is also very simple, with no widely available tools for advanced element types, localized mesh refinement, and mesh diagnostics for example.

The proposed solution is to utilize a commercial multi-physics code, with a robust FEA solver and graphical interface for problem setups, combined with the scripting capabilities of MATLAB. This will setup a strong foundation for further work in customized topology optimization, since most research works would require only small modifications to be implemented in this program. Further, it would allow for handling of more realistic scenarios and better integration into an engineering design process. The ability to translate this research work to a commercial deployment also becomes much more straightforward, as a typical CAE user would require little training to be familiarized with the new feature. The MATLAB

scripting could also easily be replaced with other languages such as Python or C++ based on development needs or for a more distribution friendly approach.

To do this, the ANSYS mechanical software will be used as the commercial piece of the implementation. It is well known and highly popular within engineering companies and should provide a strong base with its FEA capabilities. While it currently has some topology optimization implementations of its own, these are black-box solutions, with only a few parameters that are adjustable, and the underlying code cannot be modified for new work. Therefore, the custom scripting capability of the software will be utilized to build topology optimization code from scratch. The scripting language is known as the ANSYS parametric design language (APDL) and it is based on legacy programming languages such as BASIC and FORTRAN. It has many built-in commands that are generally used to automate various parts of the design process, as well as develop more advanced simulations and available results. It allows for scalar, vector, and matrix operations natively, and has a full set of logical commands, allowing for if-then-else statements, and do-loops or while-loops.

While the entire implementation could certainly be done only in APDL, the scripting interface is little more than a text editor, containing no quality-of-life features, and little in the way of debugging or optimization capabilities. The age of the programming language itself (it was first designed for use with punch-cards) also shows, with less advanced capabilities than more modern languages. Again, integration of existing and future topology optimization research would also require more extensive modification to be used in APDL. Thus, only the graphical user-

interface and FEA solver of ANSYS mechanical will be used, with MATLAB containing the topology optimization scheme, along with the new perimeter constraint.

### **4.1.2 Structure of the Program**

Since both ANSYS and MATLAB will be used to solve the problem, the way in which these programs interact requires some explanation. First, the engineering problem is setup in the ANSYS mechanical graphical user interface. Geometry is imported into the software, material properties are specified, then meshed as per the requirements of the problem. ANSYS offers advanced mesh types and refinement techniques, with meshes from other programs also being available for import. Then, boundary conditions are added, including forces, moments, pressures, fixed supports, and other, more niche constraints. Next, the parameters for the topology optimization are set, including the volume fraction, filter radius, penalty value, maximum move parameter, slicing threshold and direction, and perimeter penalty. If the user wants to specify a certain region, or certain elements that they would like to have excluded from the topology optimization, this would also be done now. At this point, the user would simply call the ANSYS solver as they normally would, and the rest would be taken care of by an APDL script.

This script is setup to be injected into the solver before any results are calculated. Here, it takes over, writing key element and node data to text files, then specifying each element with a unique material label, so they can be easily modified based on the results of the topology optimization. To modify the element's density,

as per the material distribution scheme, each element is set to use only the proportion of its Young's modulus as specified by the volume constraint. Now, the FEA solver is called and the resultant strain energy density is calculated. With this, each element's sensitivity can be calculated and subsequently written to a text file in the working directory. The user inputted parameters, the working directory, and the current iteration number are then passed along to a compiled MATLAB executable. The APDL script then waits for MATLAB to finish calculating the new element densities by looking for a text file containing their values to be written. Once it sees this, the new densities are read in, and a new FEA is solved. This process repeats until the convergence criteria is met. The user can then view the resultant geometry directly in ANSYS, along with any other result they are interested in, including stress, strain, and displacement.

When the MATLAB executable is called, it receives the parameters passed to it directly from ANSYS, and then reads in the element, node, and sensitivity information from the current iteration located in the ANSYS working directory. From here, it uses this information to slice the ANSYS mesh directly, then calculates which elements lie on the boundary contours. Based on the applied AM perimeter penalty, these element sensitivities will then be modified. This is then passed into a standard SIMP implementation, where the values are filtered, and the new densities are calculated. These values are then written into a text file to let the APDL script know it is complete. The current MATLAB execution is then terminated until it is called for the next iteration. A visual representation of this entire process is shown in Figure 4-1.

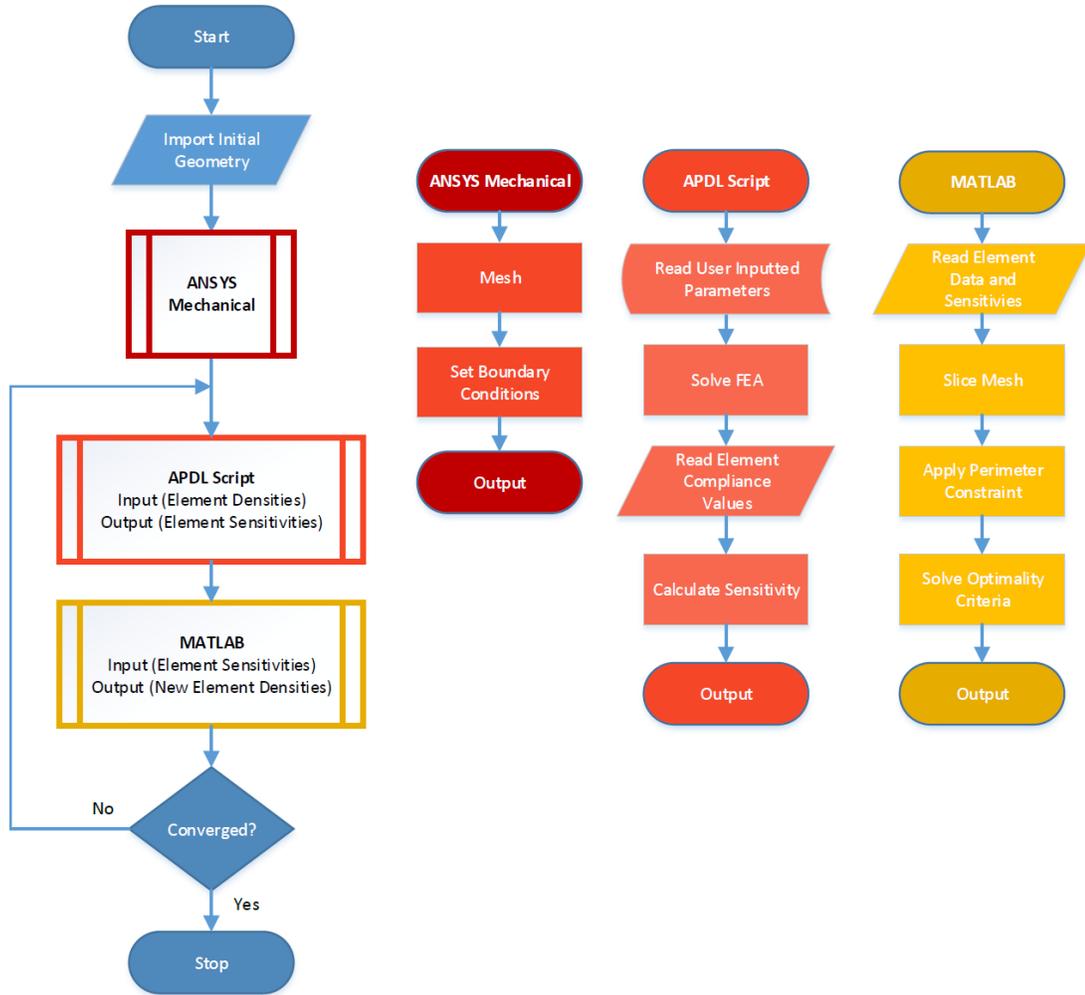


Figure 4-1. A simplified flowchart of the program interactions and logic.

## 4.2 Additive Manufacturing Perimeter Constraint

This section will detail the numerical implementation of the SIMP method, as well as the slicer and perimeter constraint as introduced in the methodology.

### 4.2.1 SIMP Implementation

As discussed previously, the major reason behind using both ANSYS and MATLAB for the implementation is the prevalence of topology optimization code in the

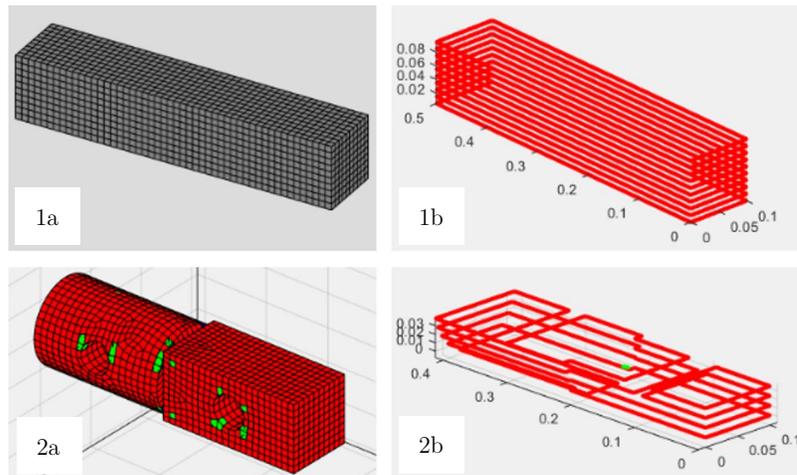
MATLAB programming language. Therefore, for this thesis, the popular 99-line code developed by Sigmund [92] will be adapted for use. Specifically, the optimality criteria used for updating element densities will be directly utilized. The remaining portions of the 99-line code are used for setting up the problem and solving the FEA, so they will not be used. The mesh filter will also not be used. This is due to how the design domain and mesh are defined and inputted for direct use in the code. Instead, a different approach to filtering is taken, but with similar end results.

To eliminate the checkerboard effect, as well as help with mesh independency, a nearest neighbour search will be conducted for the filter. Here, with the element centroids known, the distance from each element to every other element is calculated. Any element within the radius specified by the user is kept, with the others being eliminated. Then, based on the distance to each of these elements, a weighting factor is applied. To reduce the computational cost of this procedure, this calculation is performed during the first iteration only, and the list of elements within the radius and their respective weightings are written to a MATLAB variable in the working directory.

### **4.2.2 Slicer Implementation**

Now that the basic topology optimization has been detailed, the perimeter constraint must be applied. The first step is to find the elements that lay on the boundary contours. Here, a state-of-the-art FEA slicer by Bender et al. [61] is adapted. Since the code was also created in MATLAB, the implementation is fairly straightforward. First the element and nodal information are formatted and passed to the script,

allowing it to differentiate between the various element types used in ANSYS. Since the slicer was designed to only do one pre-defined slice at a time, it was modified in order to accommodate multiple slices. The mesh is sliced based on the average size of an element, using the maximum and minimum nodal heights in the direction of slicing to evenly space out the slices (Figure 4-2). Finally, the slicer returns a list of edge coordinates that form contours, labelled as either inner or outer. Since the program only returns a list of coordinates, further code was added to make this useful for the constraint application.



**Figure 4-2.** Example of meshes (1a & 2a) and their respective slices (1b & 2b).

Using the given list of edges, the total perimeter of the slice is calculated based on the summation of the edge lengths. For reference, the area of the slice is also calculated. Next, based the coordinates given, a search was done in order to determine which elements correspond to any given edge. Normally this is two elements; one that lies on the inside of the contour, and one that lies on the outside. In a near converged design, one element will be solid, and the other void. It should

be noted that these elements will both be a part of the same “global” slice inner or outer boundary contours, and thus they are not differentiated for the method presented here. Also, in situations where the slice falls directly between elements, an edge may match with 3 or 4 elements in total. This element information is then passed into two variables; one containing a list of elements that lie on the “global” outer contours, and one containing elements on the “global” inner contours.

### **4.2.3 Perimeter Constraint Implementation**

With the two lists of elements that lie on the boundary contours, the perimeter constraint can then be applied. By looping through every element in the list of sensitivities, the elements can be checked if they lie on the outer or inner contour. If they do, then the corresponding scheme in Equation 3-20 is applied, using the user inputted AM perimeter penalty value.

# Chapter 5

## Results & Validation

Various 2D case examples will be presented using the new additive manufacturing topology optimization perimeter constraint. The geometry resulting from various AM perimeter penalty values will be shown, along with the corresponding normalized perimeter, surface area, and compliance. These values will also be examined over the course of the topology optimization to gain a better understanding of the penalties function. Two problems in 3D will also be demonstrated. All results are obtained from the platform described in Chapter 4. The resulting geometry will then be taken into a popular 3D printing software where print time simulations will be compared.

### 5.1 2D Cases

Since the AM topology optimization perimeter constraint utilizes a slicing scheme, it is useful to look at a scenario where only one slice is generated for the geometry. This will give a clear picture of how the perimeter penalty directly affects the design.

Here, three examples will be detailed. Each case will be made of a square mesh with equal element sizes for simplicity. For all cases, the AM perimeter penalty,  $p_{AM}$  will be tested at 1 (off), 1.1, 1.3, 1.5, 1.7, and 1.9, and all other values set as:

$$\text{Volume Fraction} = 0.4$$

$$\text{Penalty Factor} = 2.8$$

$$\text{Filter Radius} = 1.25$$

$$\text{Convergence Criteria} = 0.1\%$$

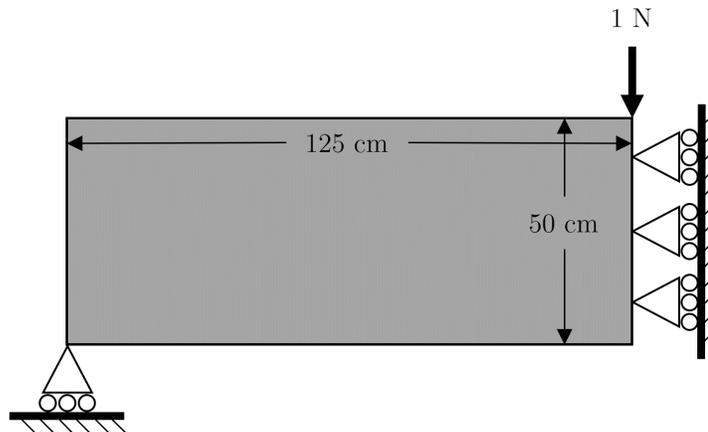
$$\text{Move Parameter} = 0.3$$

$$\text{Slicer Density Threshold} = 0.3$$

All of these are fairly standard values for a compliance-based topology optimization, except the slicer density threshold which is an addition for the new constraint. It is set to 0.3 in order to allow for the initial densities of the elements to be picked up as solid, so the constraint can apply from the very start of the optimization.

### 5.1.1 MBB Beam

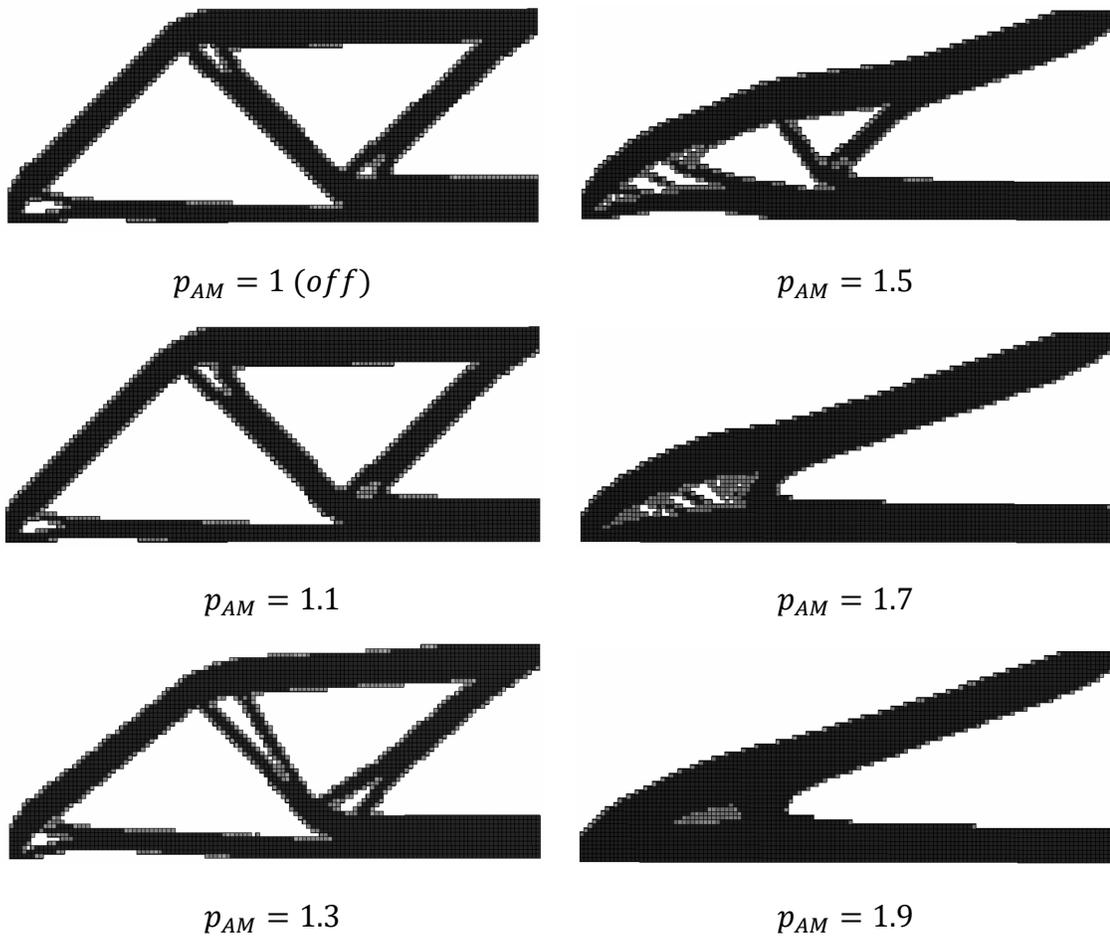
The Messerschmitt–Bölkow–Blohm (MBB) beam is an extremely popular topology optimization example. The setup is shown in Figure 5-1:



**Figure 5-1.** 2D MBB beam setup.

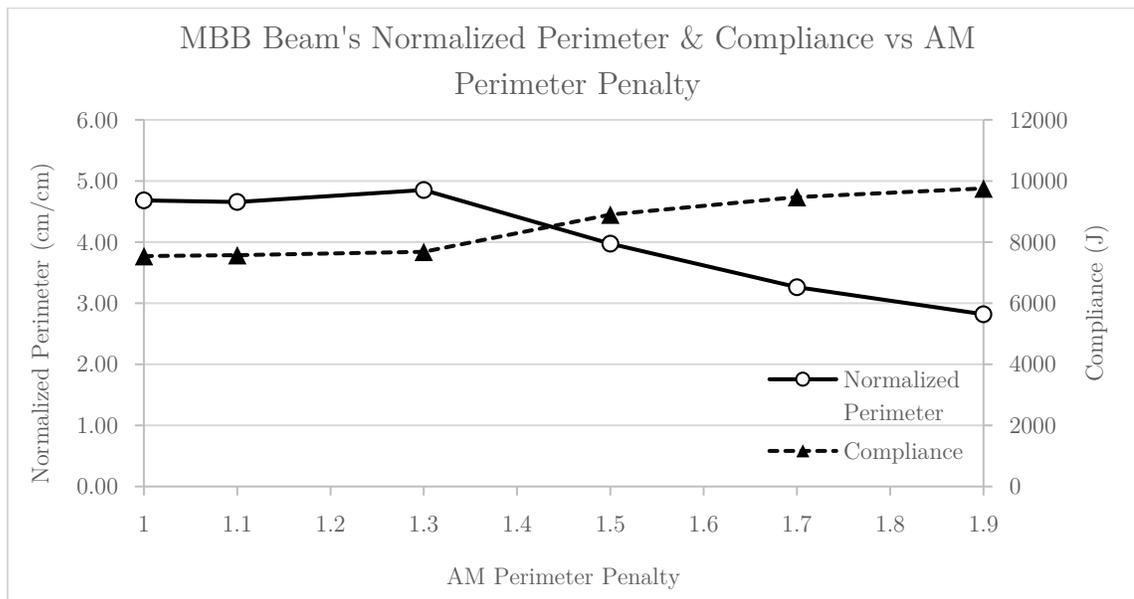
Note that the force is only 1 N due to the relative nature of topology optimization. The resulting geometry will not change if a higher force is added. Only the relative magnitude of force is needed. In this problem, the element size is set to 1 cm for a total of 6250 elements.

The resulting geometry can be seen in Figure 5-2. Values higher than  $p_{AM} = 1.9$  do not converge. Note that as the AM perimeter penalty increases, the mass of the beam seems to concentrate further. The lighter patches of elements also show that the optimization did not achieve a fully 0-1 design. However, it should be noted that the slicer is set to consider all elements above 0.3 density as solid.



**Figure 5-2.** Resulting geometry for the MBB beam.

Next, Figure 5-3 shows a plot of the change in compliance and normalized perimeter based on the applied AM perimeter penalty value. It can be seen that the perimeter decreases proportionally once a high enough penalty is reached. In this case,  $p_{AM} = 1.5$ . Correspondingly, the compliance increases, showing a loss in stiffness. Compared to the result without any applied penalty, the maximum applied penalty,  $p_{AM} = 1.9$ , sees a 39.8% reduction in normalized perimeter, and an increase of 29.3% in compliance. The surface area stays very similar across each penalty value, with a reduction of 2.9% for  $p_{AM} = 1.9$  compared to no penalty.



**Figure 5-3.** Change in the MBB beam’s compliance and normalized perimeter based on AM perimeter penalty.

Next, in Figures 5-4 and 5-5 the change in normalized perimeter and compliance are detailed over the convergence of the optimization for each penalty value. For clarity, every 10<sup>th</sup> iteration is shown. Geometry at noteworthy iterations is also pictured, with a gradient of solid (black) to void (white). Here it can be seen that as the penalty value increases, the number of iterations to converge generally

increases. For each penalty, a trend of increasing perimeter that levels off and then decreases is noted. A special case for penalties 1.7 and 1.9 exists where the perimeter increases significantly, then stagnates for a period, and finally decreases. This, along with the greater number of iterations to converge, could possibly be explained by a lengthened existence of intermediate (gray) densities caused by the aggressiveness of the penalty. In fact, when penalties above 1.9 are applied, the optimization never exits this stage and therefore never converges. The compliance follows the inverse of this trend, where it decreases rapidly due to the formation of stiff solid members, then stabilizes and only increases when the penalty value is high enough.

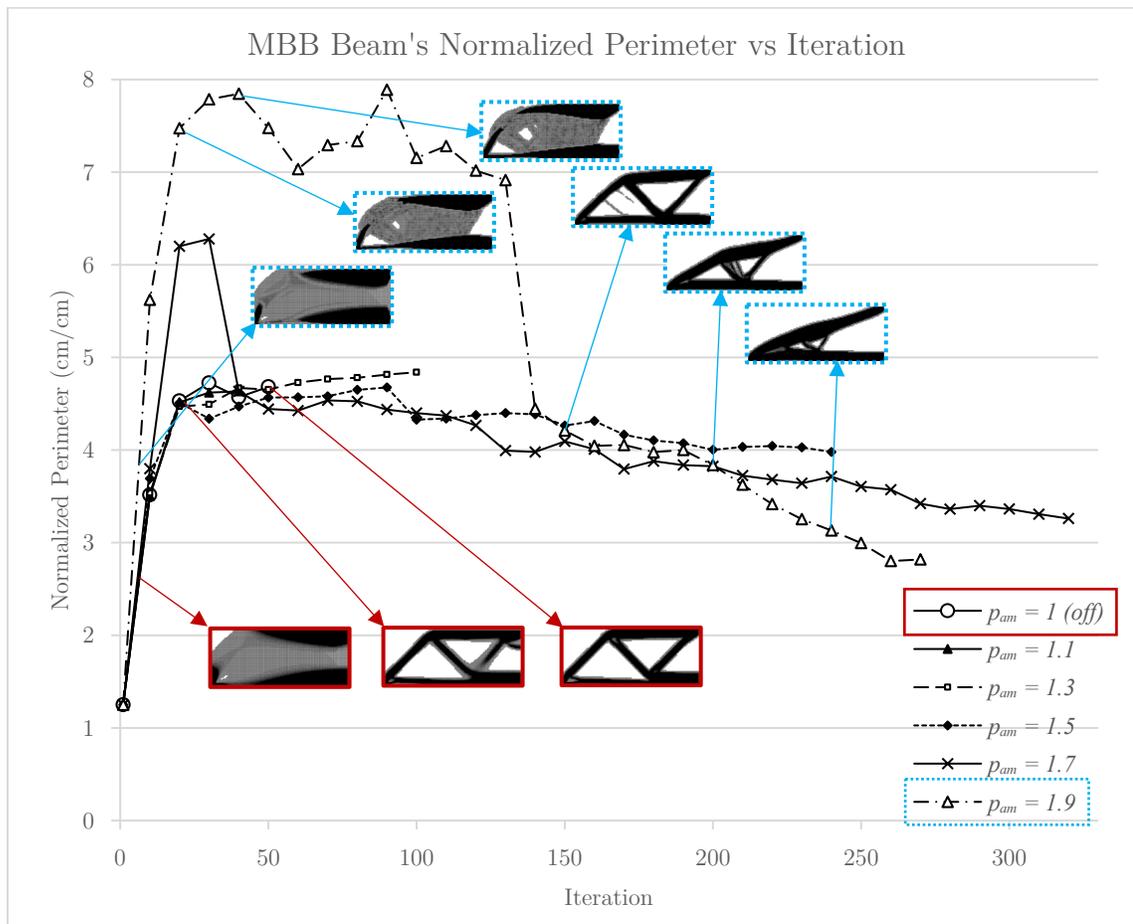
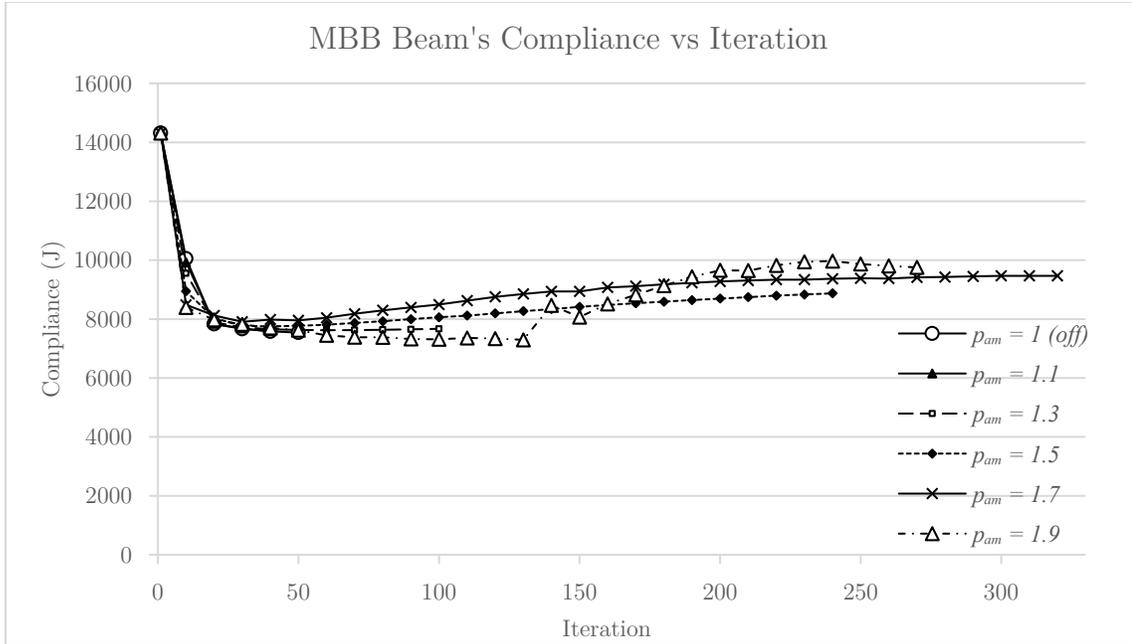


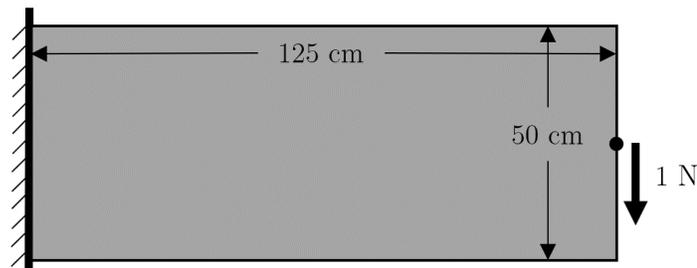
Figure 5-4. Change in the MBB beam's normalized perimeter over each iteration.



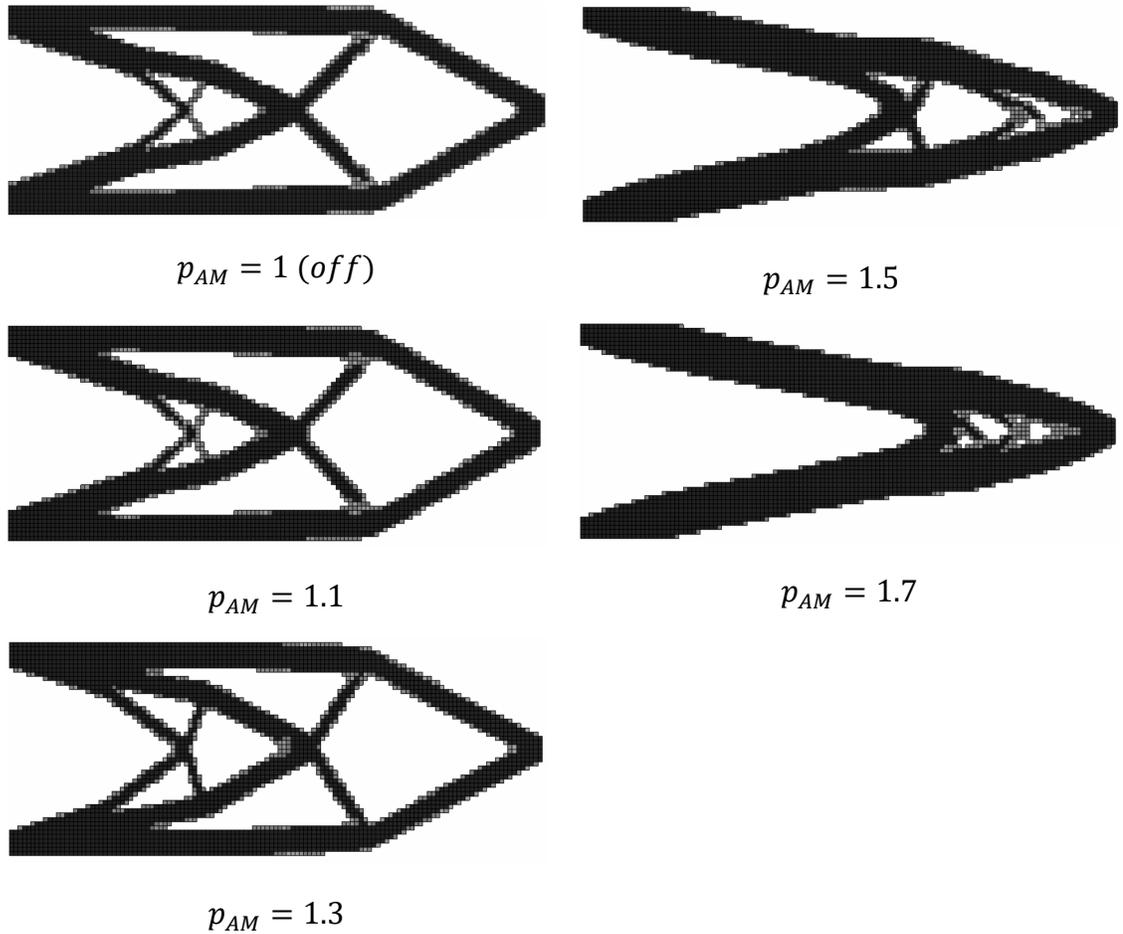
**Figure 5-5.** Change in the MBB beam's compliance over each iteration.

### 5.1.2 Cantilever Beam

Another popular example to examine is the cantilever beam (Figure 5-6). The element size is set to 1 cm for a total of 6250 elements. The resulting geometry can be seen in Figure 5-7. Here, values higher than  $p_{AM} = 1.7$  do not converge. Again, the same mass concentration effect is seen, along with some intermediate densities remaining.

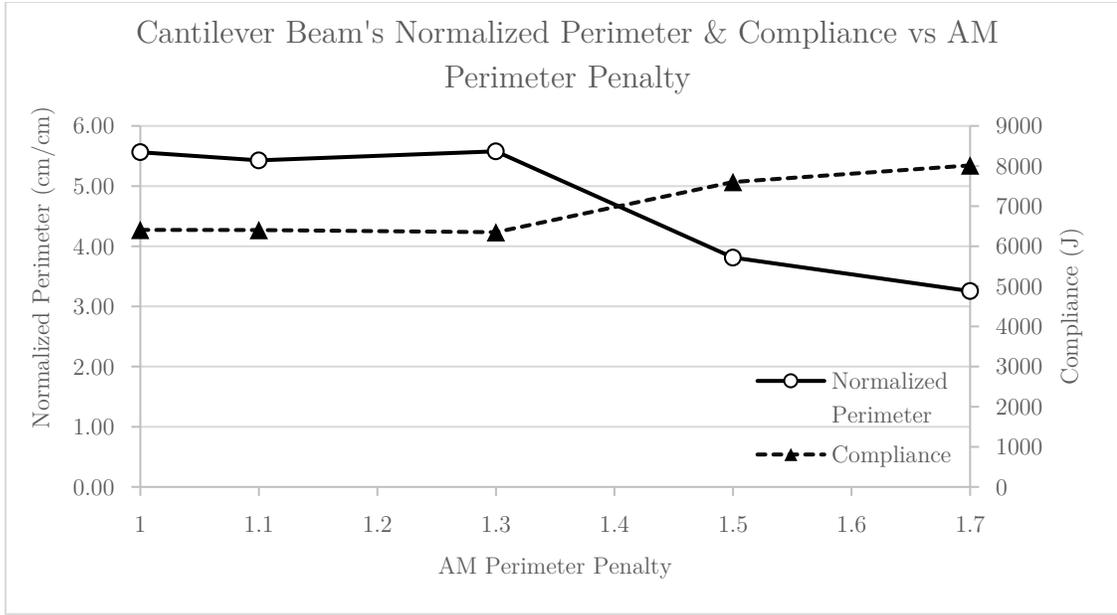


**Figure 5-6.** 2D cantilever beam setup.

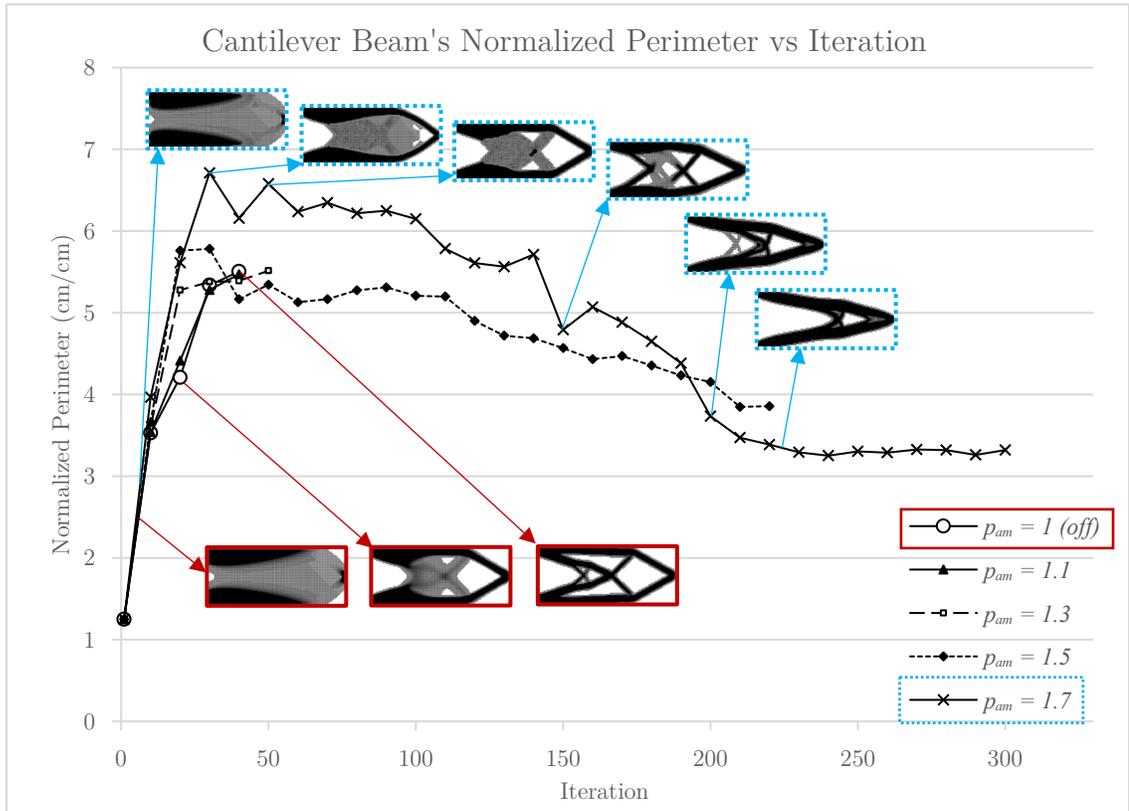


**Figure 5-7.** Resulting geometry for the cantilever beam.

Figure 5-8 shows the change in compliance and normalized perimeter compared to the penalty value. The same trends from section 5.1.1 also exist here. The normalized perimeter decreased by 35.1% and the compliance increased by 25.1% for the maximum applied penalty as compared to no perimeter constraint. The surface area also holds steady for each penalty value, with a 3.7% reduction for the maximum penalty. Figures 5-9 and 5-10 show each iterations normalized perimeter and compliance. The trends are similar to the MBB beam, however less of the special case of stagnation is seen, with only  $p_{AM} = 1.7$  showing some of the effect.



**Figure 5-8.** Change in the cantilever’s compliance and normalized perimeter based on AM perimeter penalty.



**Figure 5-9.** Change in the cantilever’s normalized perimeter over each iteration.

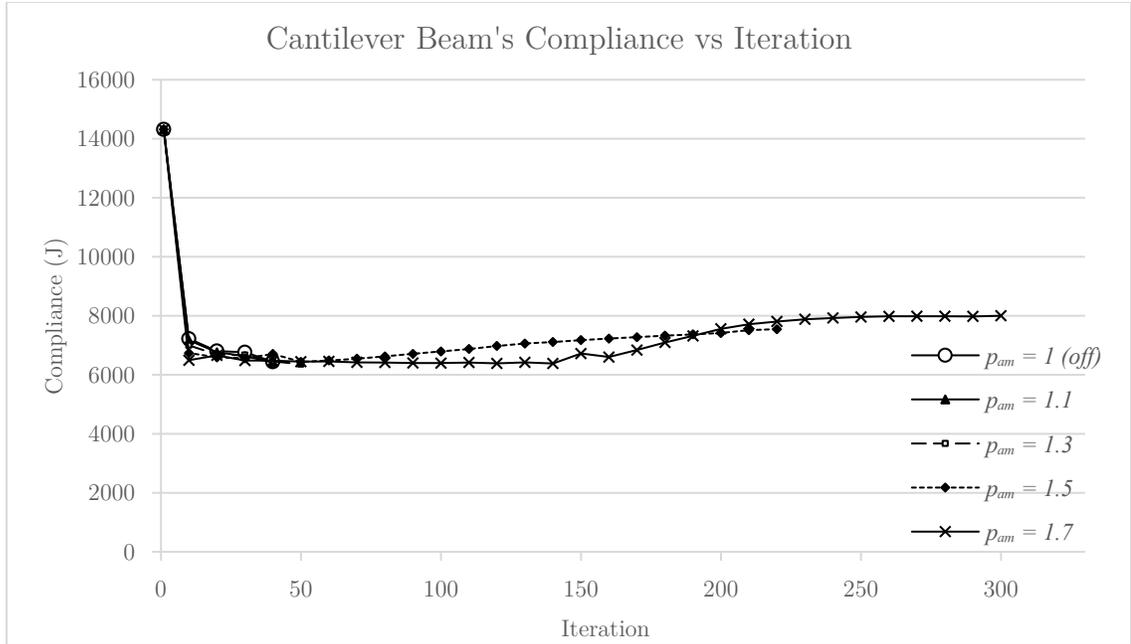


Figure 5-10. Change in the cantilever's compliance over each iteration.

### 5.1.3 Multiple Load Case Bridge

A simple bridge setup with two loads is pictured in Figure 5-11. A total of 6750 elements are used, each with a size of 1 cm. For this setup, penalty values higher than 1.7 do not converge. The resulting geometry can be seen in Figure 5-12. The mass concentration effect is seen here, along with a number of intermediate density elements.

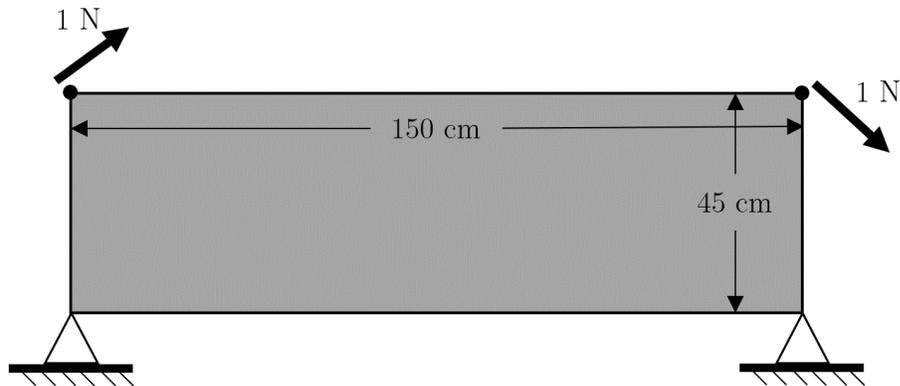
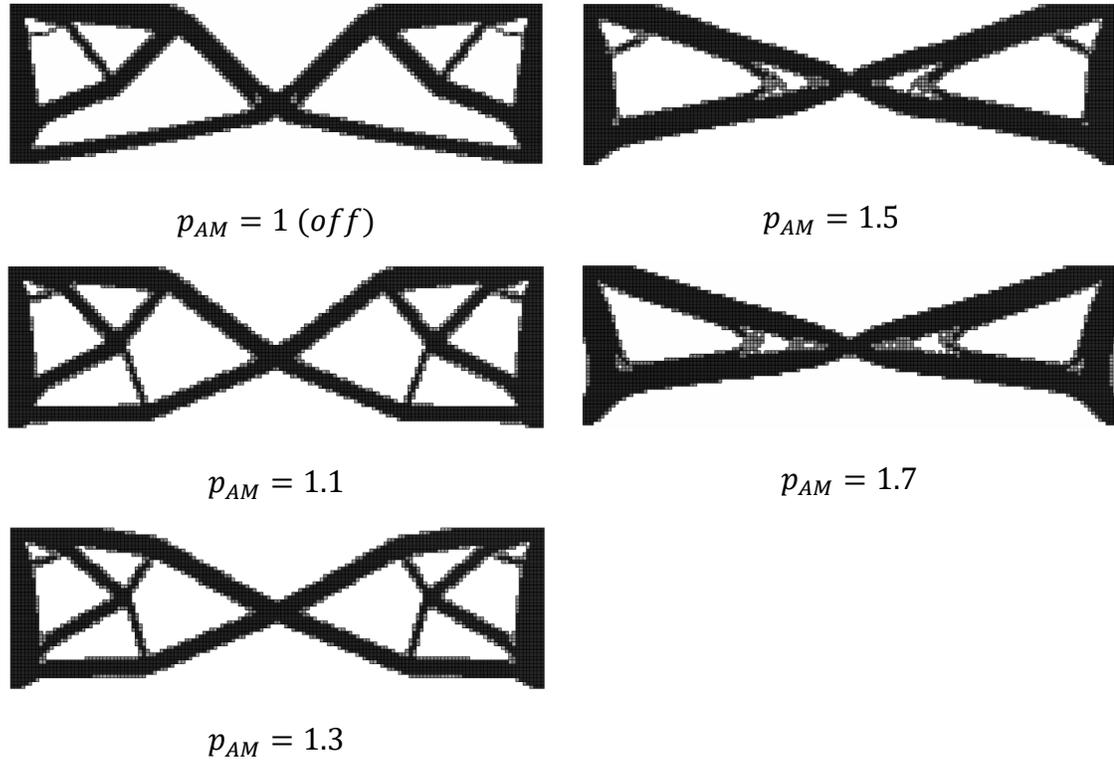
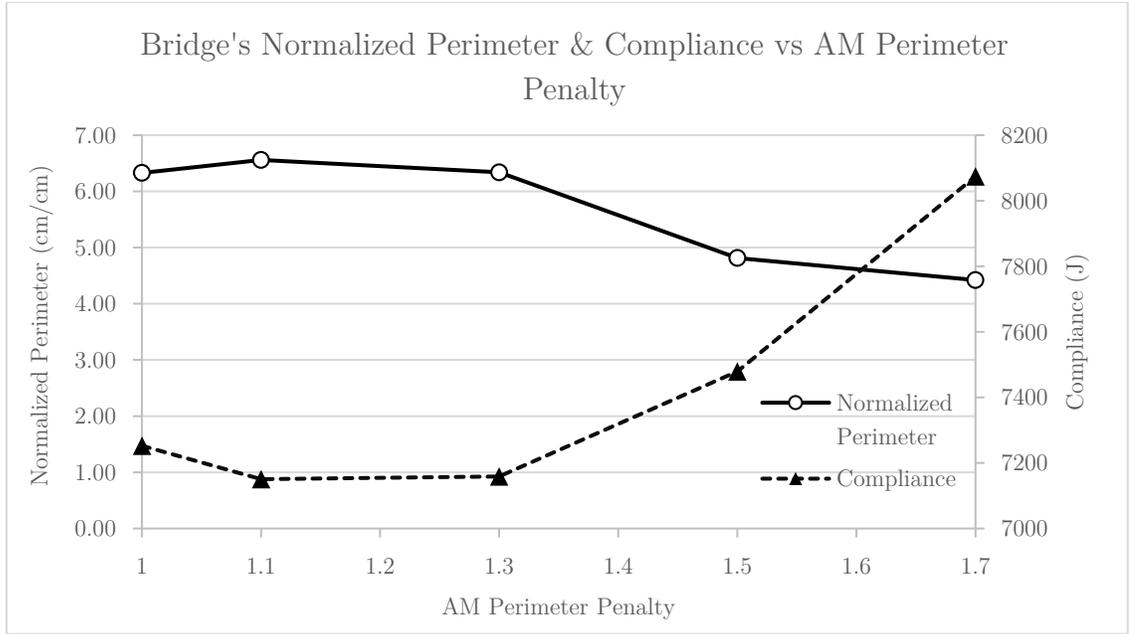


Figure 5-11. 2D bridge setup.

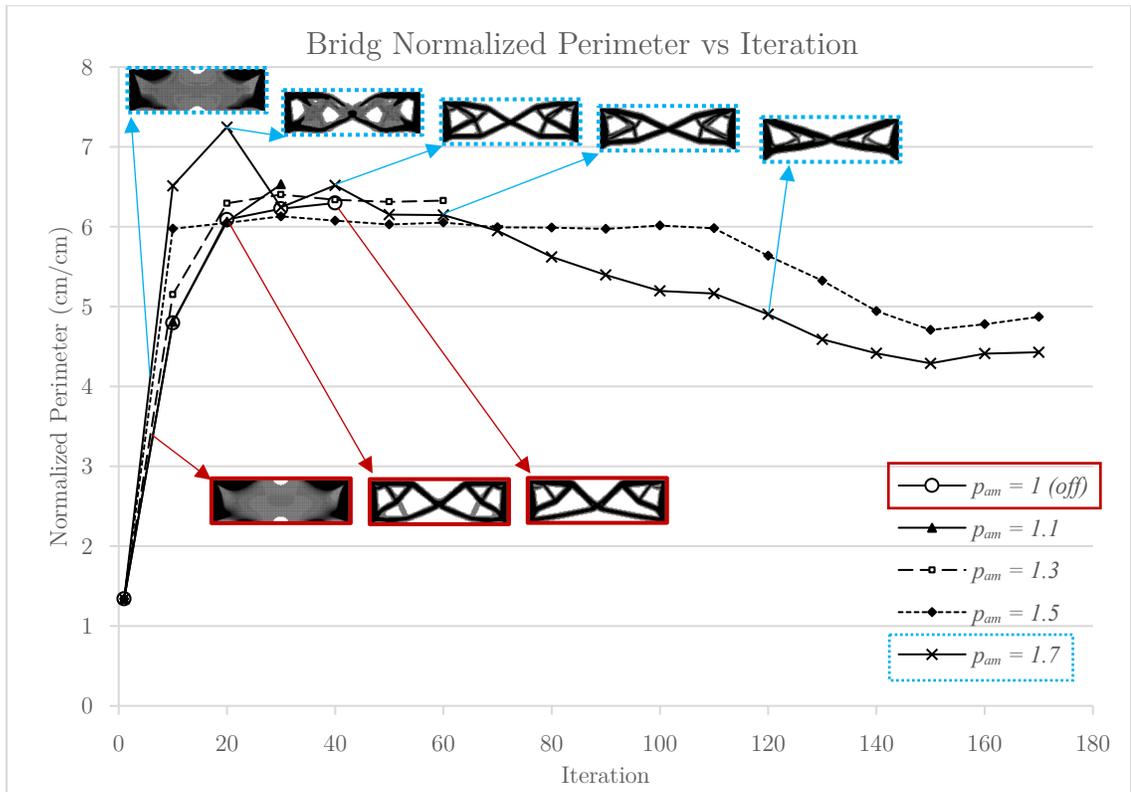


**Figure 5-12.** Resulting geometry for the bridge.

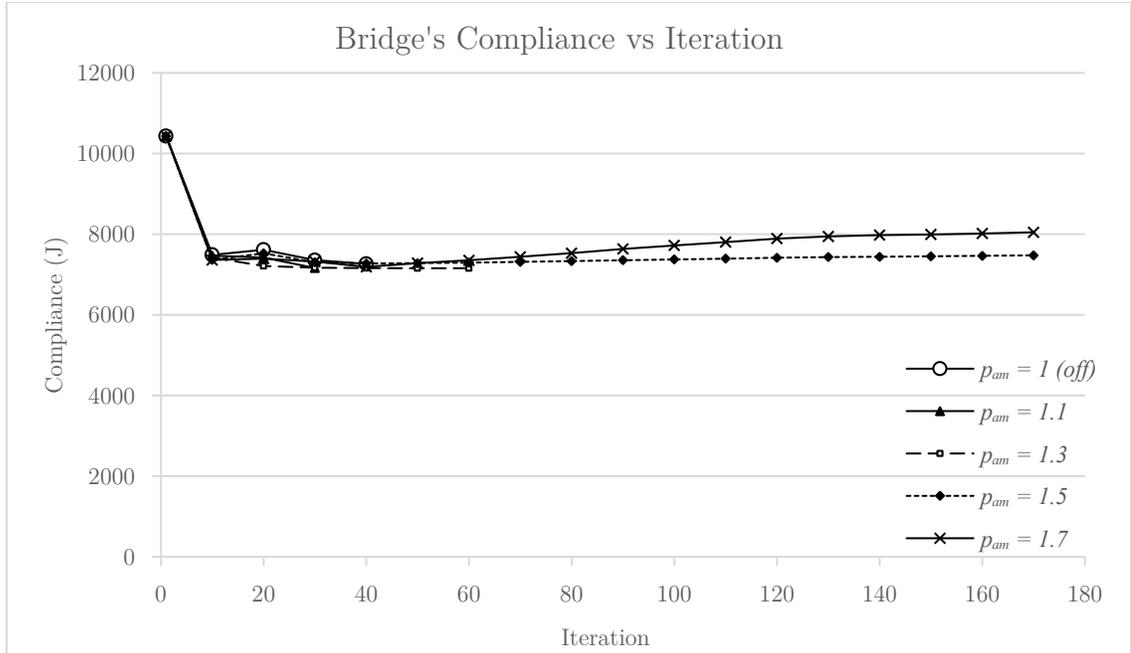
The compliance and normalized perimeter for each penalty value is shown in Figure 5-13. The trends from the previous two sections are seen here as well. For the maximum penalty applied, a decrease of 30.1% for the normalized perimeter, and an increase of 11.3% for the compliance is seen compared with no penalty. Each iterations normalized perimeter and compliance are detailed in Figures 5-14 and 5-15. Trends from the previous sections are seen as well, except the special case of a large increase in perimeter followed by a stagnation period. Penalty  $p_{AM} = 1.7$  is the closest to following this feature due to a slightly larger increase in perimeter compared to the other penalty values. However, it does not demonstrate stagnation, rather, it quickly decreases to follow along with the other values.



**Figure 5-13.** Change in the bridge’s compliance and normalized perimeter based on AM perimeter penalty.



**Figure 5-14.** Change in the bridge’s normalized perimeter over each iteration.



**Figure 5-15.** Change in the bridge's compliance over each iteration.

## 5.2 3D Cases

Now, the full implementation in 3D will be presented with two examples. Slice height will be set to an evenly divided cubic mesh. The AM perimeter penalty will be tested at 1 (off), 1.3, 1.5, 1.7, and 1.9 for both cases. All other values will be set as:

$$\text{Volume Fraction} = 0.4$$

$$\text{Penalty Factor} = 2.8$$

$$\text{Filter Radius} = 1.25$$

$$\text{Convergence Criteria} = 0.1\%$$

$$\text{Move Parameter} = 0.3$$

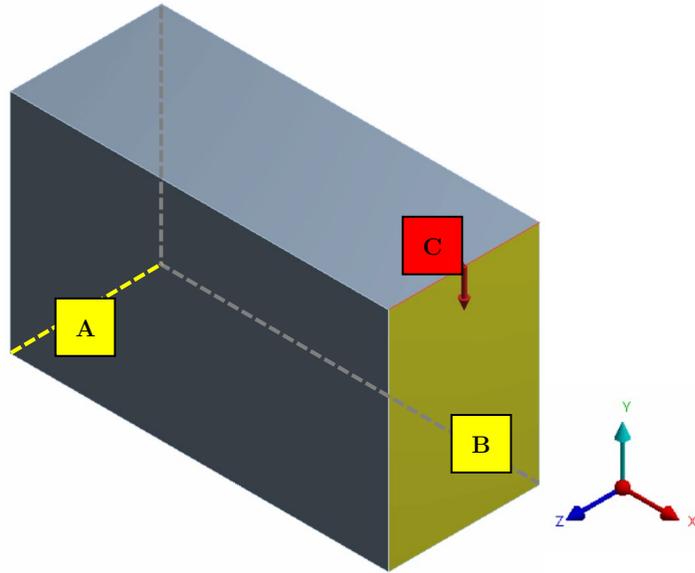
$$\text{Slicer Density Threshold} = 0.3$$

$$\text{Slice Height} = 1 \text{ cm}$$

The slicer density value is set for the same reason mentioned for the 2D cases, and the slice height is set to produce one set of contours for each layer of elements.

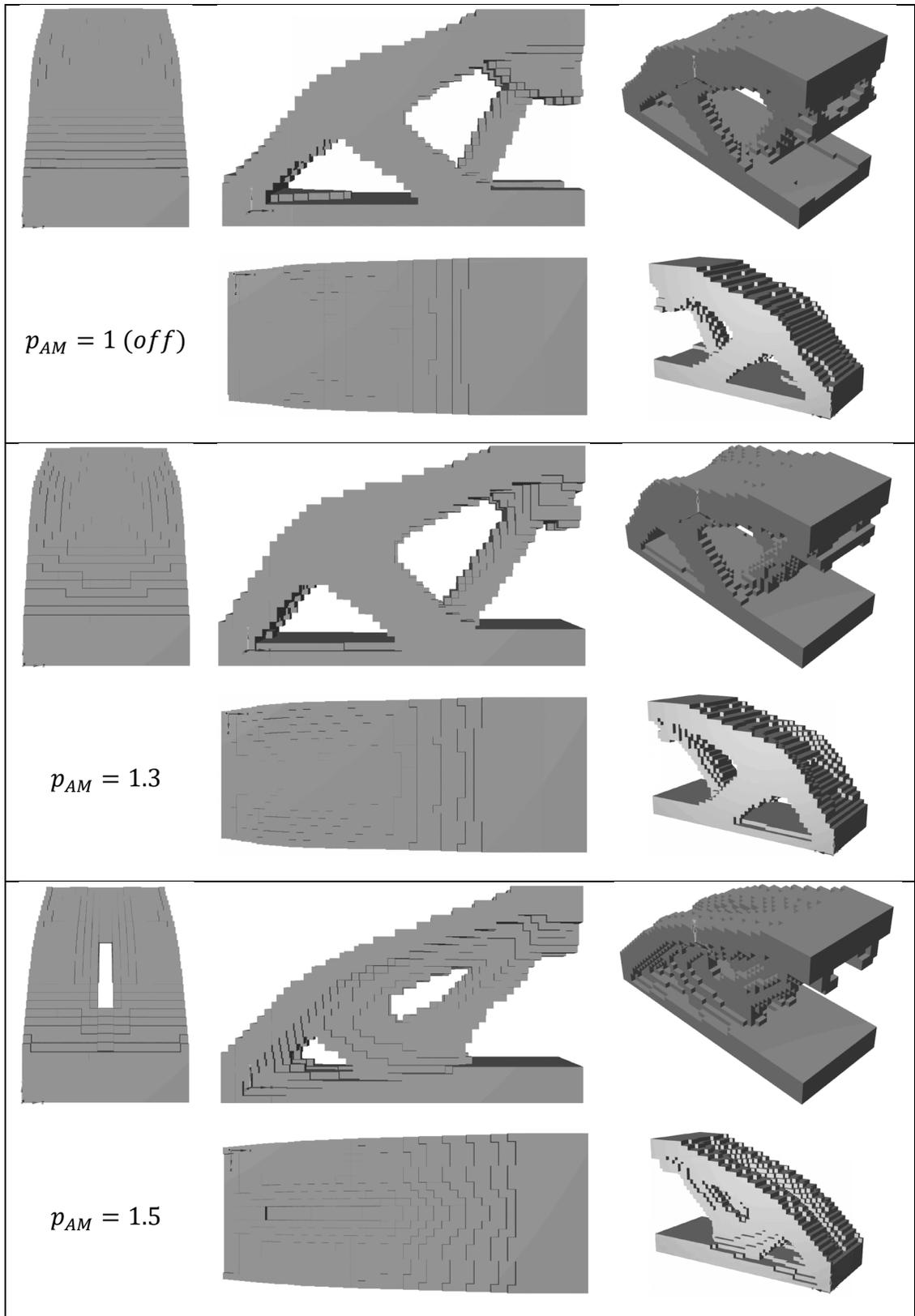
### 5.2.1 3D MBB Beam

Here, the MBB beam will be revisited except now it will be in three dimensions. The setup can be seen in Figure 5-16. A force of 1 N is applied along the top edge (C, red), with the corresponding face on rollers (B, yellow), free in the y-axis and fixed in the z-axis and x-axis. The rear edge (A, yellow) is on rollers only allowing movement in the x-axis. The slicing direction will be along the z-axis. The dimensions of the beam are 50 cm  $\times$  30 cm  $\times$  20 cm (x y z), with an element size of 1 cm for a total of 30,000 elements and 20 slices along the z-axis.

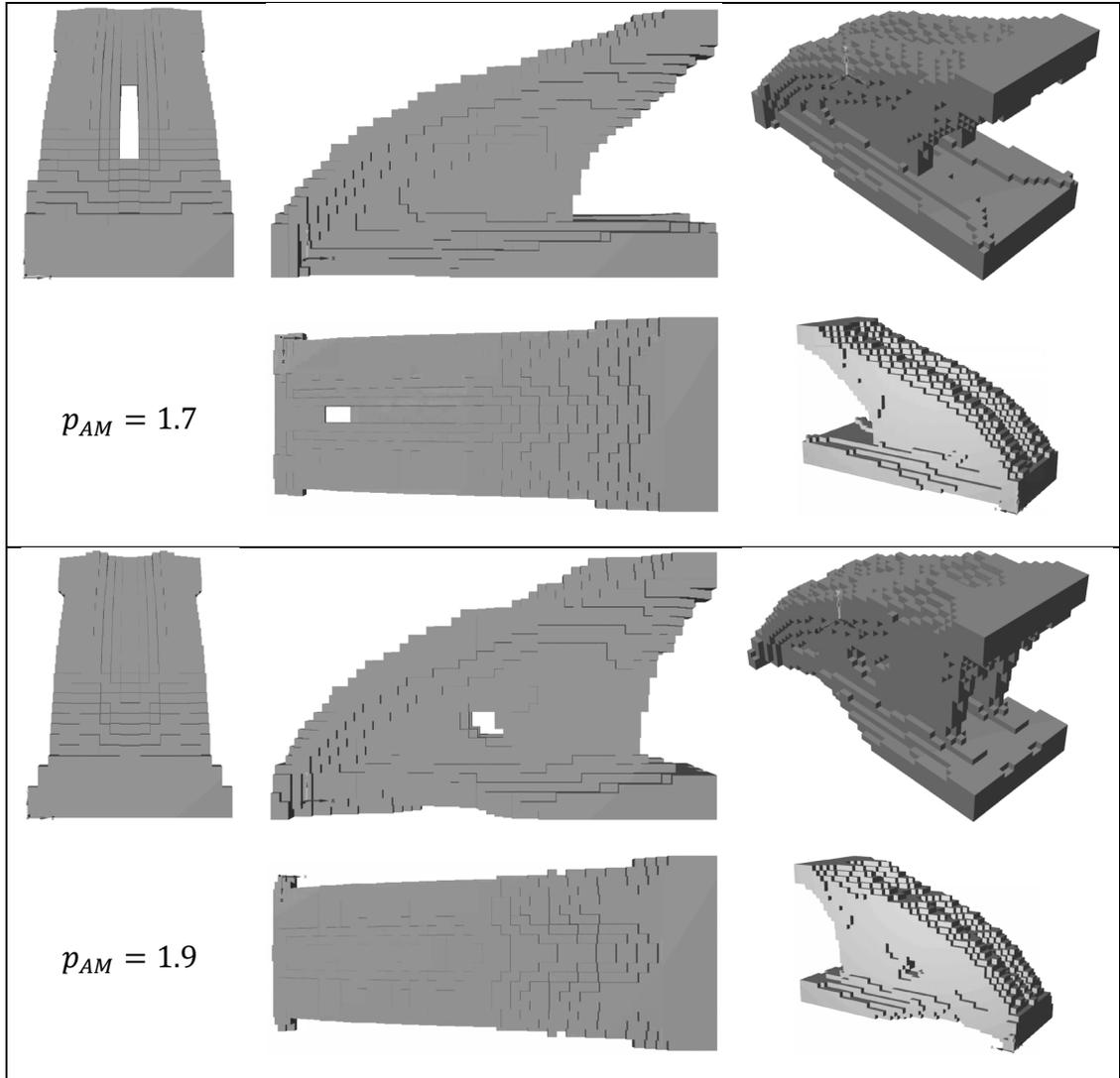


**Figure 5-16.** 3D MBB beam setup.

The resulting geometry can be seen in Figures 5-17 and 5-18. The same mass concentration results from the 2D examples can be seen here. Now, the mass also converges along the z-axis, moving away from the boundaries in addition to converging along the x and y-axis. Penalty values higher than  $p_{AM} = 1.9$  fail to solve.



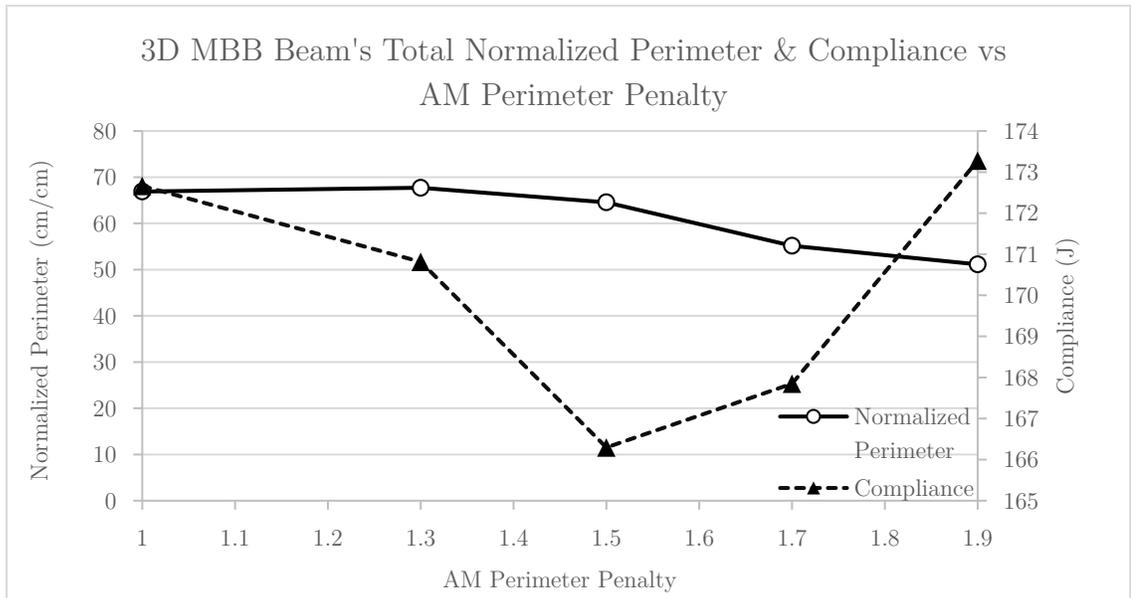
**Figure 5-17.** Part I - Resulting geometry for the 3D MBB beam.



**Figure 5-18.** Part II - Resulting geometry for the 3D MBB beam.

Since there are now multiple slices and thus perimeters corresponding to each slice, the total summation of these perimeters will be used in the following figures. The total normalized perimeter and compliance for each penalty value are shown in Figure 5-19. Even though this problem now includes multiple slices in 3D, the trend in perimeter minimization is similar to the previous 2D examples. However, the compliance trend is quite different. In the past examples, the compliance would increase along with the perimeter penalty, especially once the penalty was 1.5 or

larger. In this example, the compliance actually decreases initially, and then increases to about where it was without any penalty. This behaviour seems to point towards the initial optimization falling into a local minima, and thus when the extra constraint is added, it pulls the solution towards an even more ideal shape by chance. While the effect is positive in this solution, it is not the intended effect of the new additive manufacturing constraint, and thus this performance trend is not recognized as significant. Overall for the largest penalty value, the normalized perimeter decreased by 23.6%, with the compliance slightly decreasing by 0.4% compared to no penalty. The surface area remains steady for each penalty value, with a decrease of 0.2% seen with the largest penalty.



**Figure 5-19.** Change in the 3D MBB beam’s compliance and total normalized perimeter based on AM perimeter penalty.

The trends for normalized perimeter remain similar to the 2D cases, with the exception of the special case. Here, little to no perimeter stagnation occurs, with  $p_{AM} = 1.9$  showing only a small initial relative increase in perimeter (Figure 5-20).

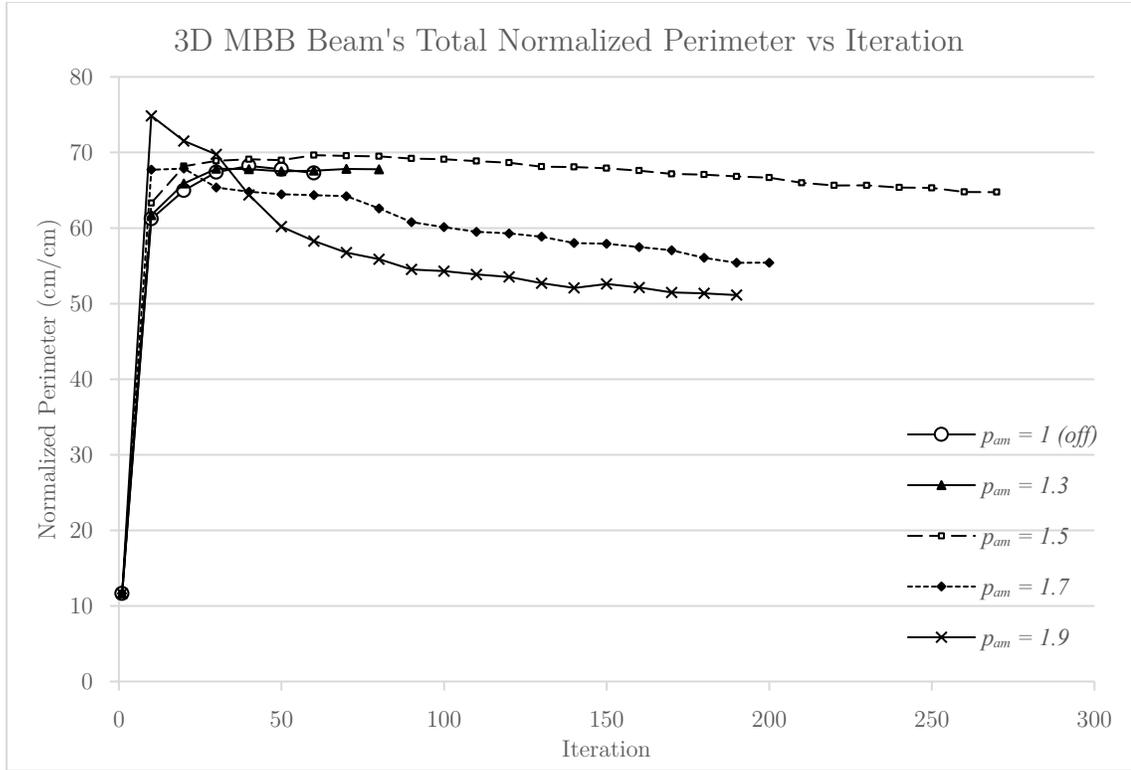


Figure 5-20. Change in the 3D beam's normalized perimeter over each iteration.

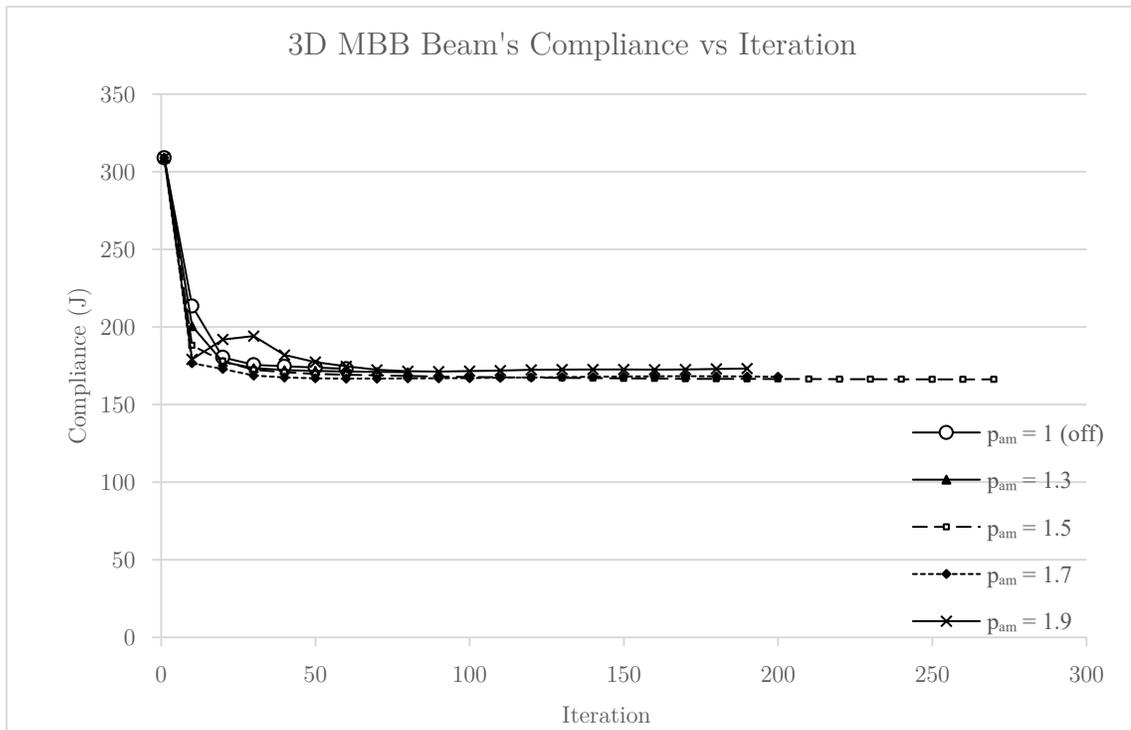
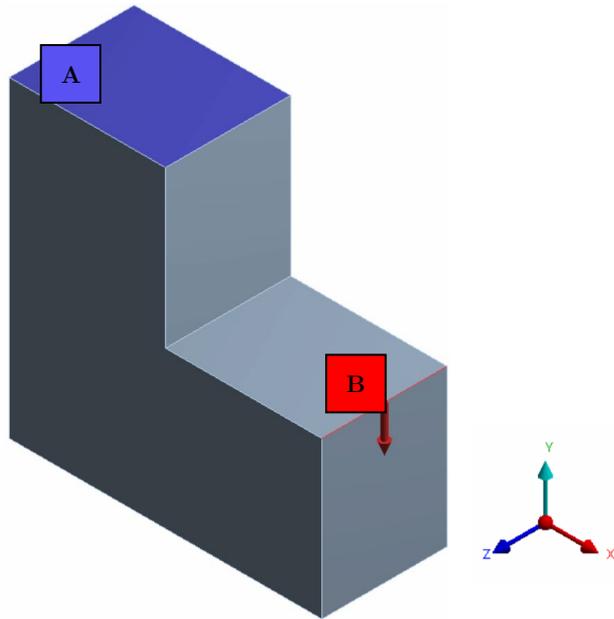


Figure 5-21. Change in the 3D beam's compliance over each iteration.

The compliance values over the optimization are then presented in Figure 5-21. The trend follows what was noted in Figure 5-19, with the compliance not increasing as the iterations increase.

### 5.2.2 3D L-Bracket

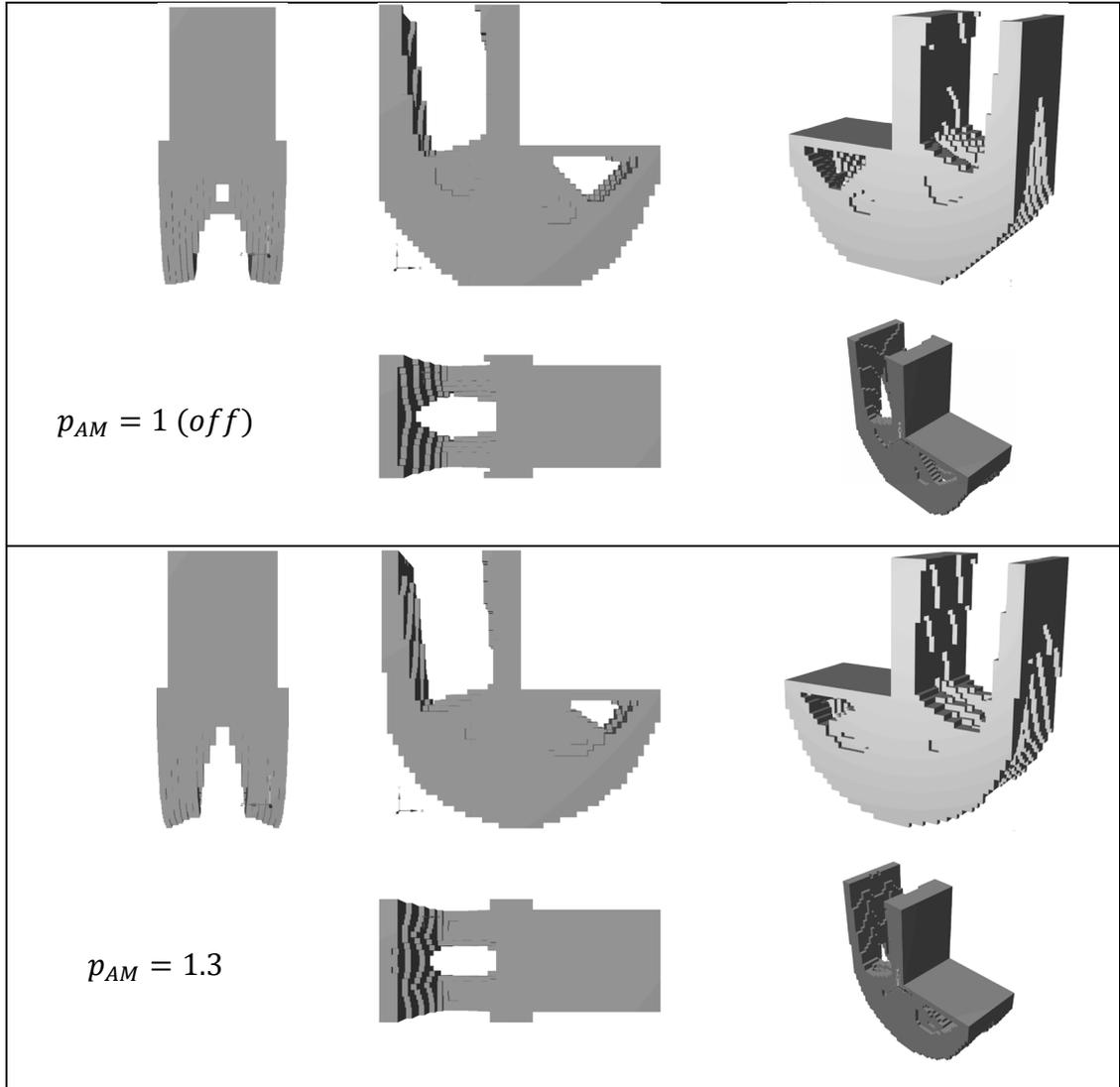
The final case will examine an L-bracket in 3D. The setup can be seen in Figure 5-22 where a downward force is applied along the top edge of the bottom step (B, red) with the blue portion (A) on the top face acting as a fixed constraint:



**Figure 5-22.** 3D L-bracket beam setup.

The dimensions are 50 cm  $\times$  50 cm  $\times$  20 cm, with a 25 cm  $\times$  25 cm cut-out as pictured. There is a total of 37,500 elements, each 1 cm in size. The slicing direction is along the z-axis for a total of 20 slices. The resulting geometry is shown in Figure 5-23 and Figure 5-24. Mass concentration can be seen once again, with the bottom

of the “L-shape” particularly moving upwards and squeezing along the z-axis. I can be noted however, that the change is not as great as in previous examples, with only  $p_{AM} = 1.9$  showing significant changes. Values above this do not converge.



**Figure 5-23.** Part I - Resulting geometry for the 3D L-bracket.

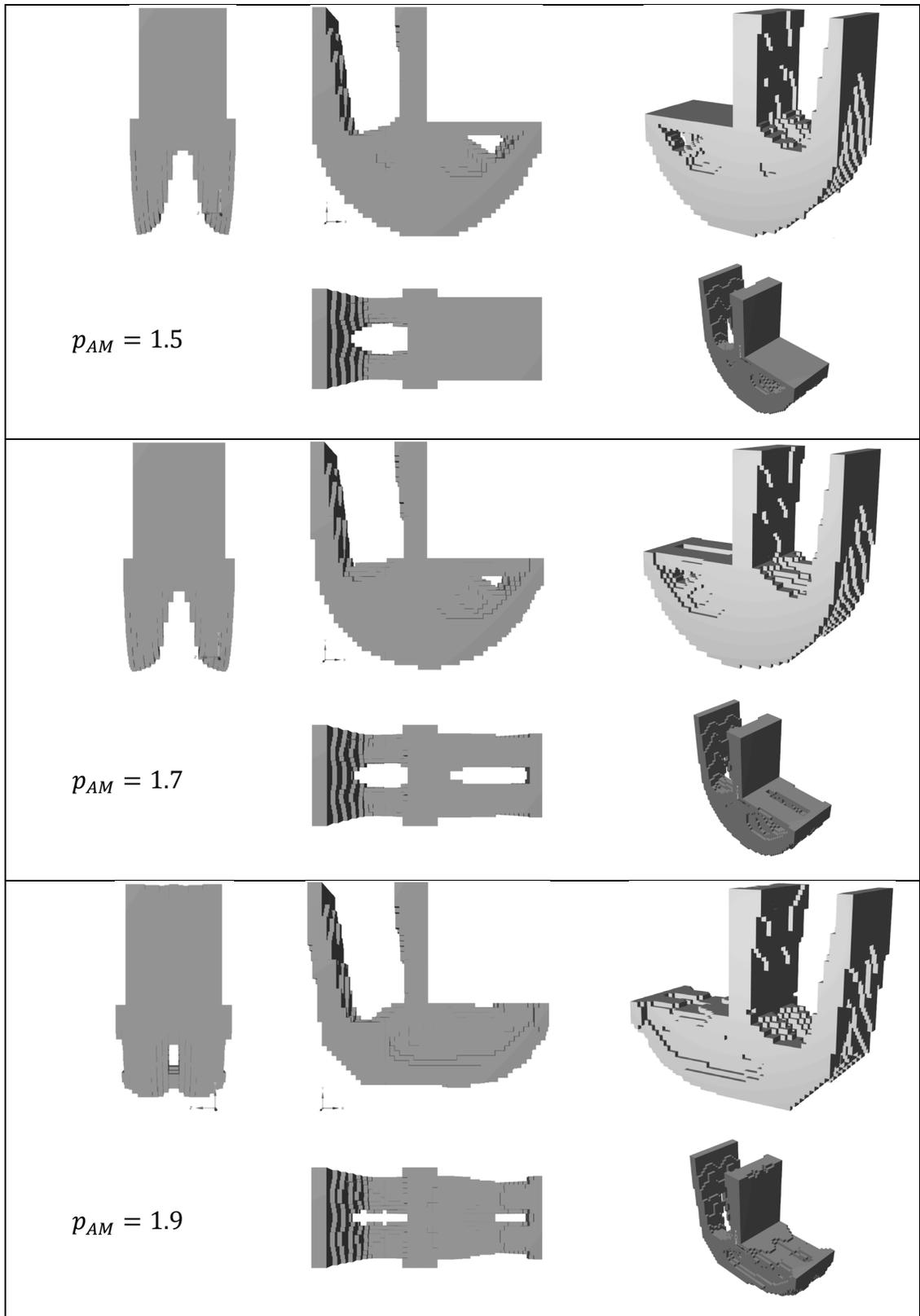
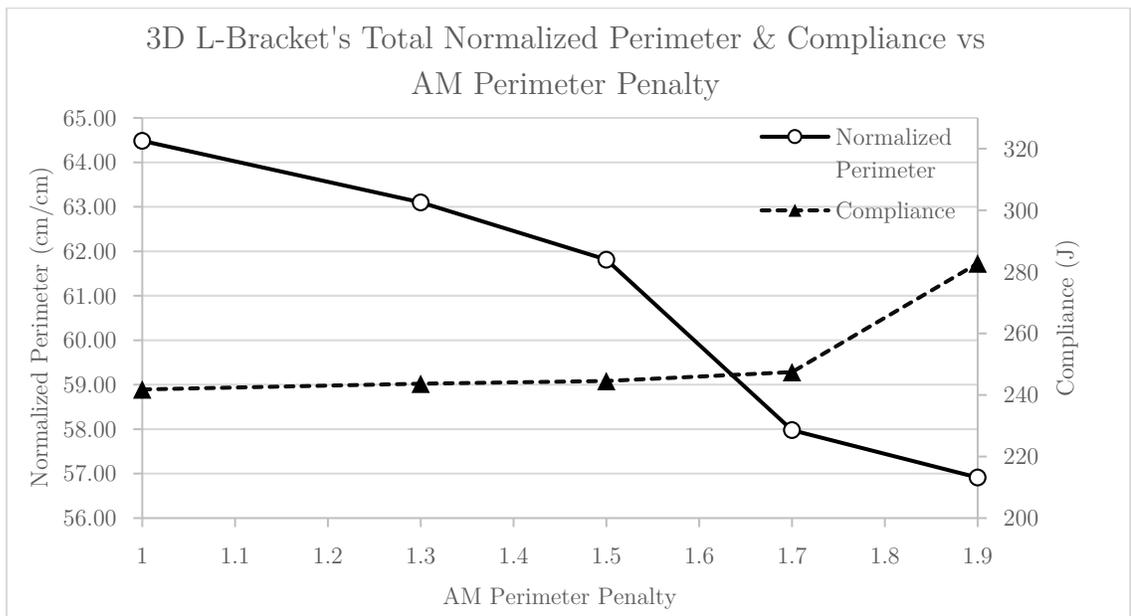


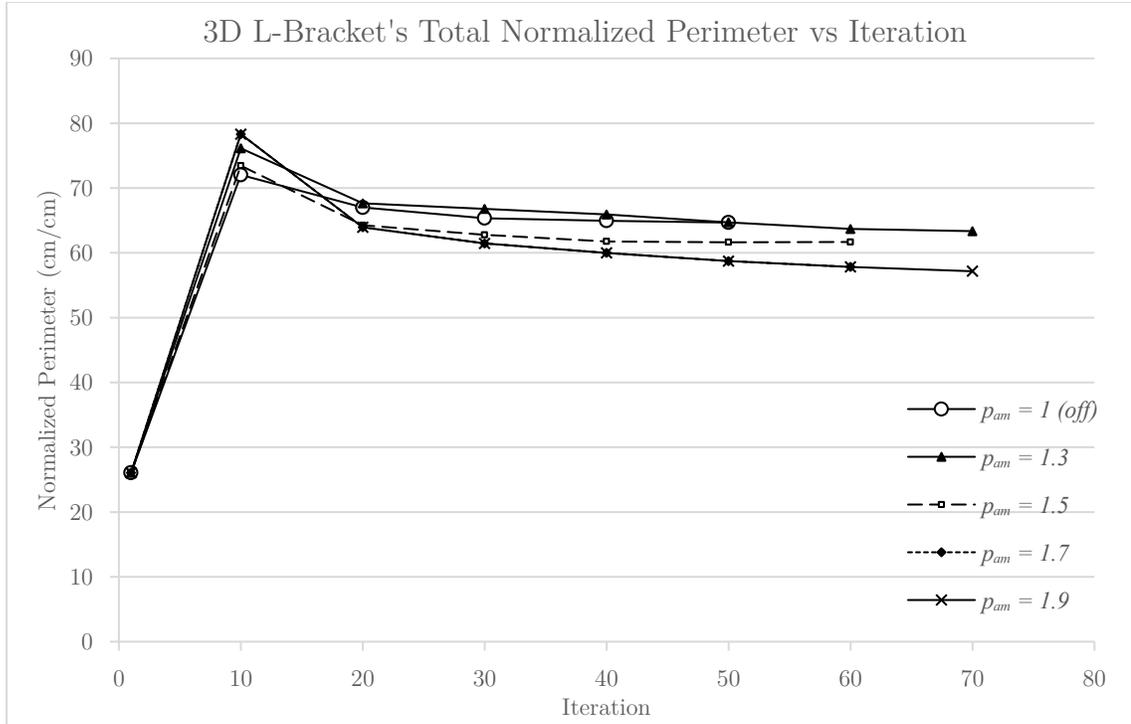
Figure 5-24. Part II - Resulting geometry for the 3D L-bracket.

Next, the total normalized perimeter and compliance are presented for each penalty value (Figure 5-25). The trend of reducing perimeter is seen here, and unlike the previous example, the compliance values look very similar to the 2D examples. This gives more credence to the idea that the previous example was trapped in a local minima for the initial optimization, and by chance found a more ideal solution with the new constraint. This example has more modest gains, with a maximum normalized perimeter reduction of 10.8% at the highest penalty value, with a 16.9% increase in compliance and 0.1% increase in surface area.

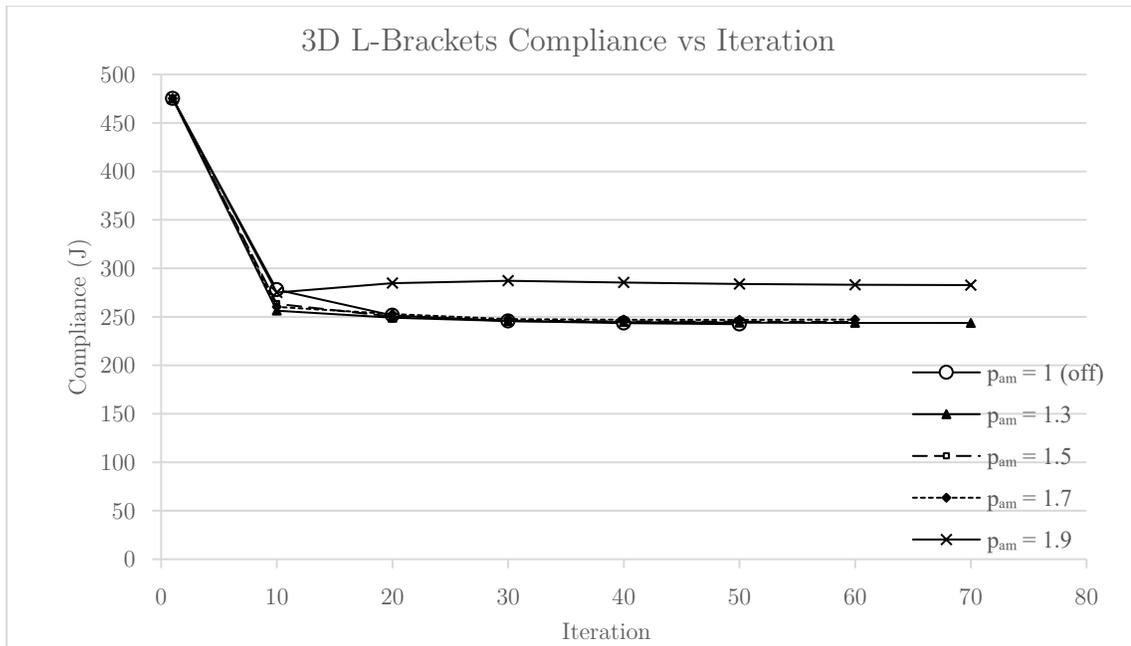


**Figure 5-25.** Change in the 3D L-bracket's compliance and total normalized perimeter based on AM perimeter penalty.

Figures 5-26 and 5-27 detail the change in total normalized perimeter and compliance over each iteration. Trends follow those of the previous examples with no special case of stagnating perimeter. It is also noteworthy that all penalty values show a similar number of iterations to converge.



**Figure 5-26.** Change in the 3D L-bracket's total normalized perimeters over each iteration (for clarity only every 10<sup>th</sup> result is plotted).



**Figure 5-27.** Change in the 3D L-bracket's compliance over each iteration (for clarity only every 10th result is plotted).

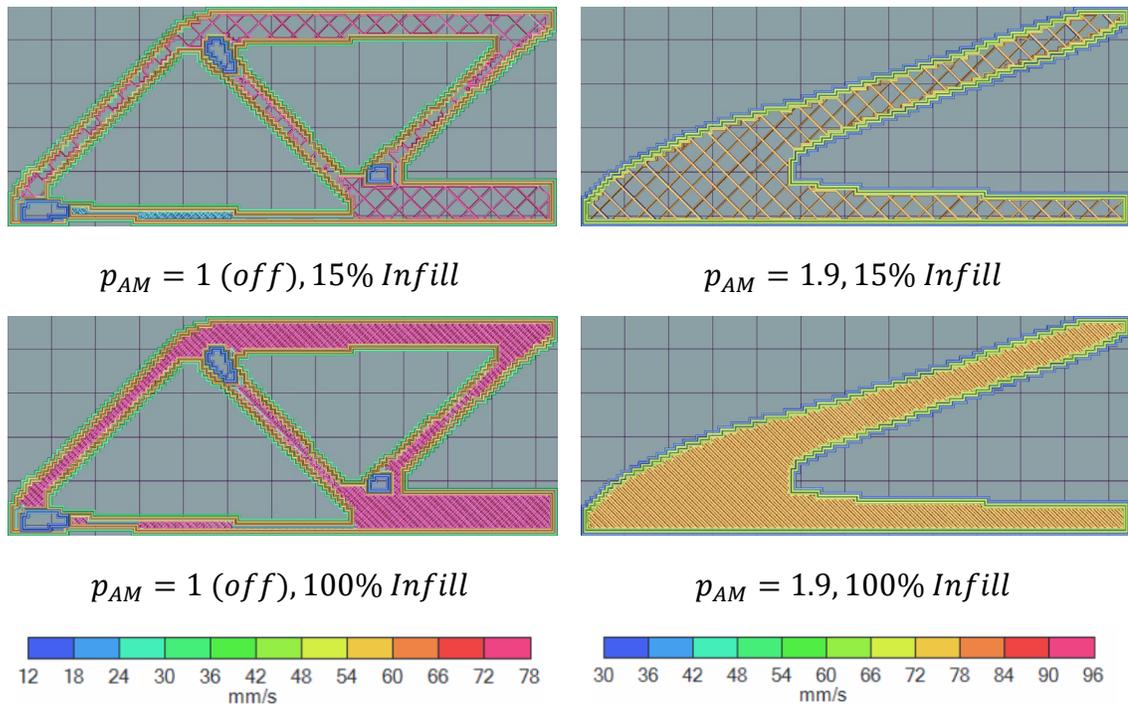
### 5.3 Validation of Print Time Reduction

Now that multiple cases have been presented to demonstrate the effect of the AM topology optimization perimeter constraint, it must be seen if the original assumption that this will translate to print time savings holds true. To do this, the open-source 3D printer software Repetier-Host and slicing software Slic3r will be used to estimate print time. The final geometries will be exported as STL files for compatibility with the software, with elements above 0.3 density considered as solid to match what was chosen for the slicer density threshold. Toolpaths will then be generated using the software, and the print time will be estimated from this.

This method uses the generated g-code and machine parameters to provide the estimate. It is by no means a representation of the exact time it will take to print an object for every 3D printer, since this is highly dependent on the exact machine and settings used. However, it will give a metric to judge the relative performance of the perimeter penalty. The amount of material, or filament, consumed is also calculated. Although the new constraint is only designed to reduce print time, it is expected that material usage will be reduced with the use of infill. In this section, each case will be examined at four infill settings, 15%, 30%, 50%, and 100% using the rectilinear pattern, as these should give a good picture of the relationship between the perimeter outlines and the surface area. Nozzle diameter is set to 0.3 mm, with 3 perimeter shells. The material calculation is set using standard 1.75 mm diameter filament. To isolate the perimeter constraint from other variables, support generation is disabled, along with any other setting that would interfere with the toolpath generation (e.g., first layer differences and solid infill layers).

### 5.3.1 MBB Beam

In this example the original  $125\text{ cm} \times 50\text{ cm}$  size is scaled to  $12.5\text{ cm} \times 5\text{ cm}$  to fit on a standard 3D printer bed. Since this is a 2D example, only one layer will be printed. Figure 5-28 shows an example of some toolpaths generated by the slicer. The colours here represent the speed the extruder will be traveling relative to the build platform. Note that in these pictures the speeds are identical even though the colour scale is different. This is due to the automated scale generated by the software.



**Figure 5-28.** Resulting toolpaths for various MBB beam results with identical slicer settings (Slic3r).

Next, Figures 5-29 and 5-30 shows the simulated print time and material usage for each penalty value. It can be seen that initially the small penalty values have little, or even a negative effect on print time. However, just like the normalized perimeter values, once a high enough penalty value is reached, the print time and

material usage substantially reduce. The proportion of reduction is also linked to the amount of infill used, with lower values yielding greater reduction as seen in Figure 5-31. A maximum of 46.7% and 33.1% reduction in print time and material usage is produced for the  $p_{AM} = 1.9$  geometry using 15% infill. The material reduction approaches zero for 100% as expected due to the very similar surface areas.

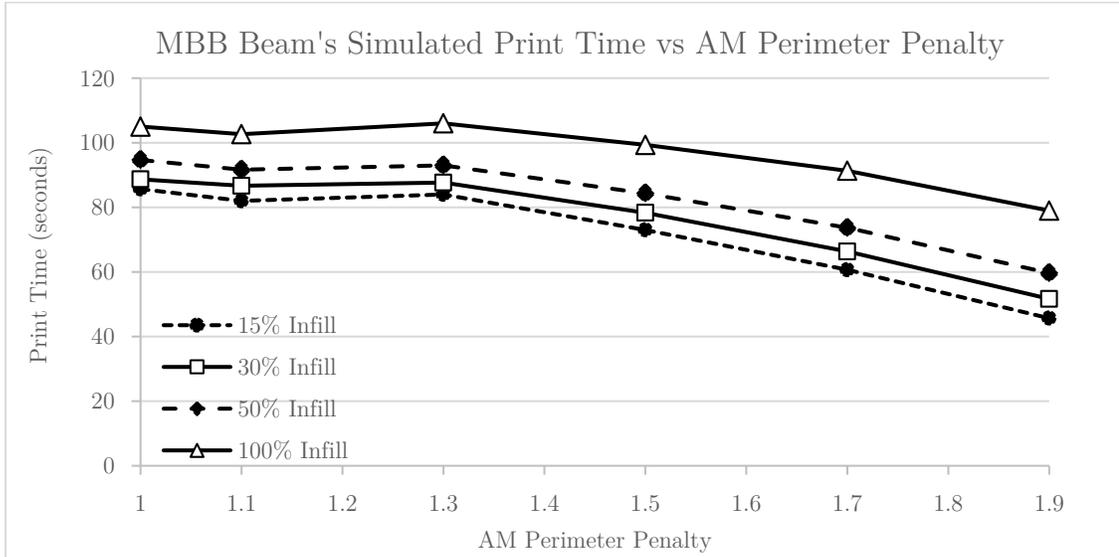


Figure 5-29. Change in the MBB beam’s simulated print time.

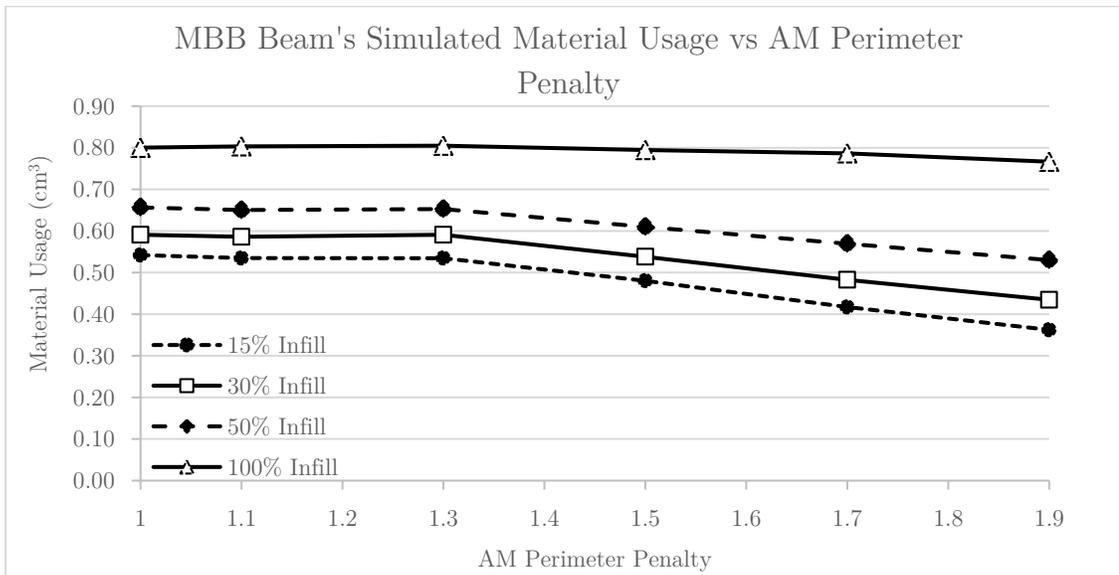
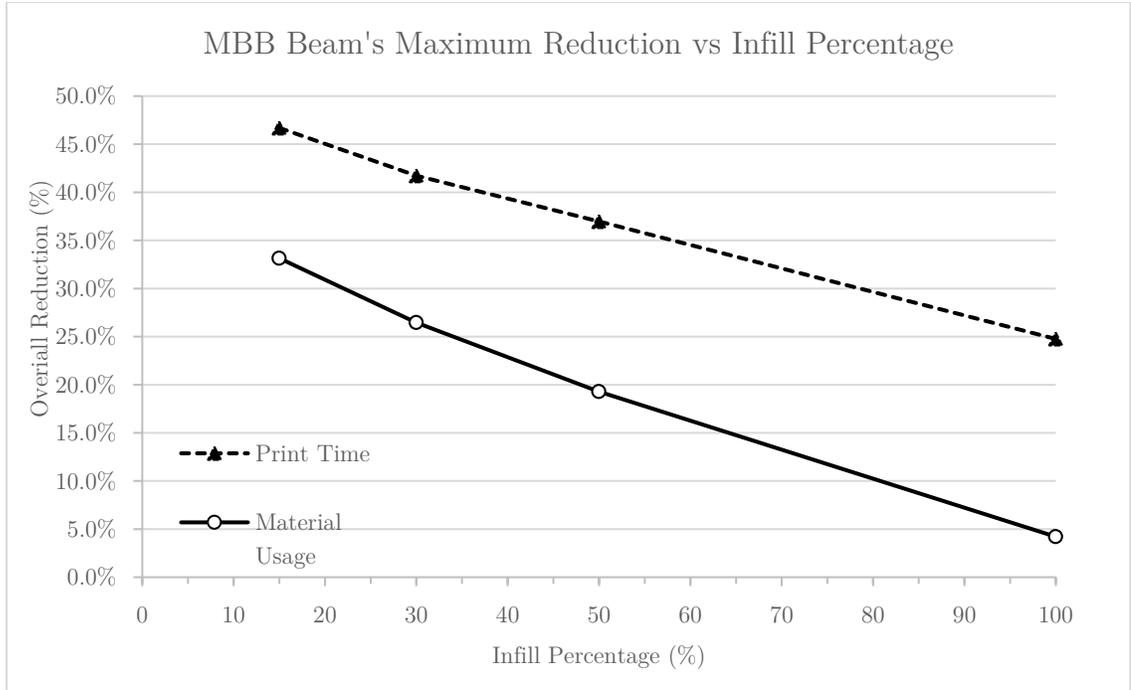


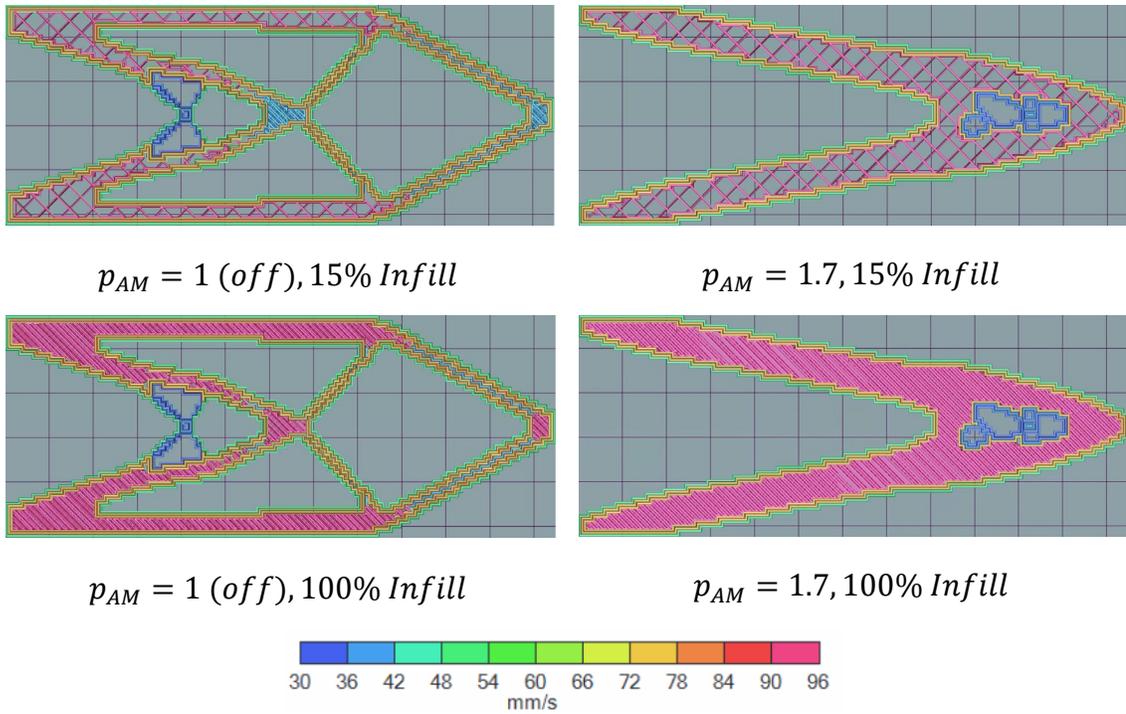
Figure 5-30. Change in the MBB beam’s simulated material usage.



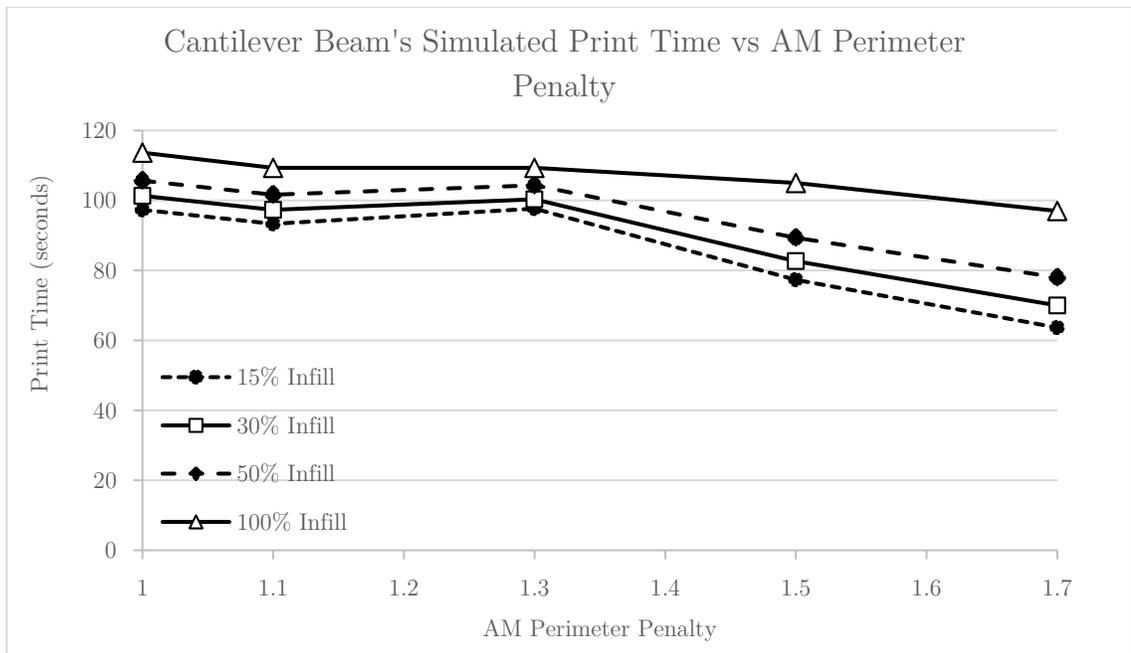
**Figure 5-31.** Change in the MBB beam’s simulated print time and material usage based on the selected infill percentage for  $p_{AM} = 1.9$ .

### 5.3.2 Cantilever Beam

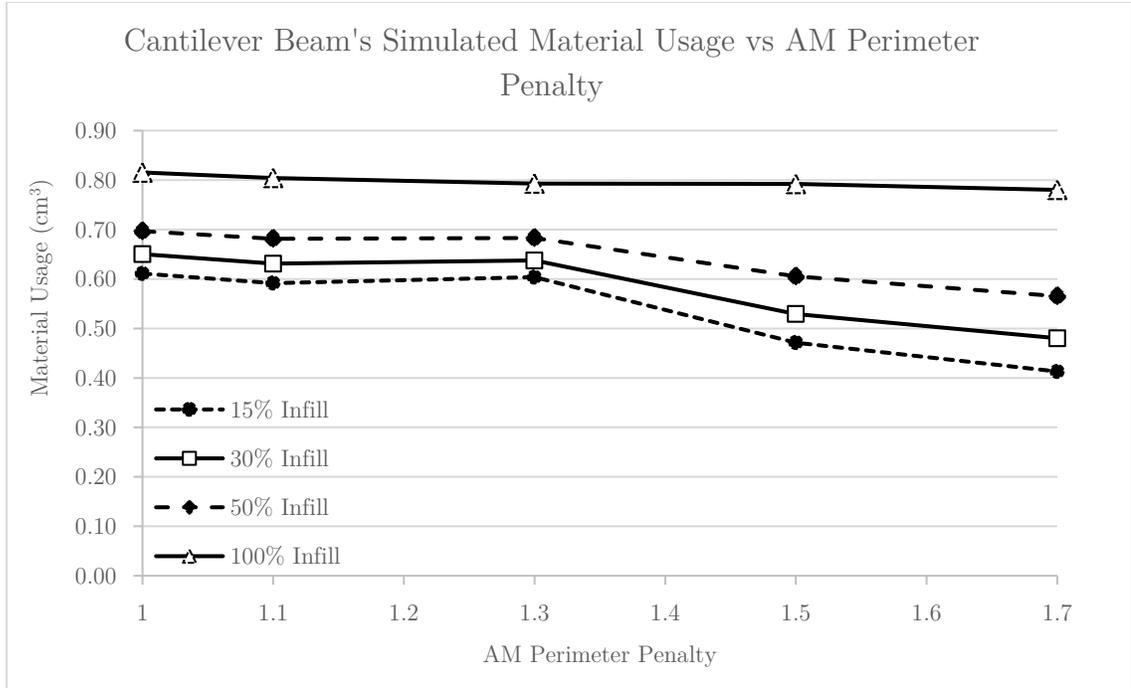
To fit a standard 3D printer bed, the design domain of the original geometry is scaled to 12.5 cm × 5 cm. Again, only one layer is printed due to the 2D profile. Some of the generated toolpaths are pictured by Figure 5-32. Here, any colour differences are due to rendering errors in the printing visualization. Speeds between the different shapes are identical. The simulated print time and material usage are presented in Figures 5-33 and 5-34. The trends seen in the previous example are noted as well. The maximum reductions for each infill selection are shown in Figure 5-35, with 34.6% and 32.4% being the largest for print time and material usage.



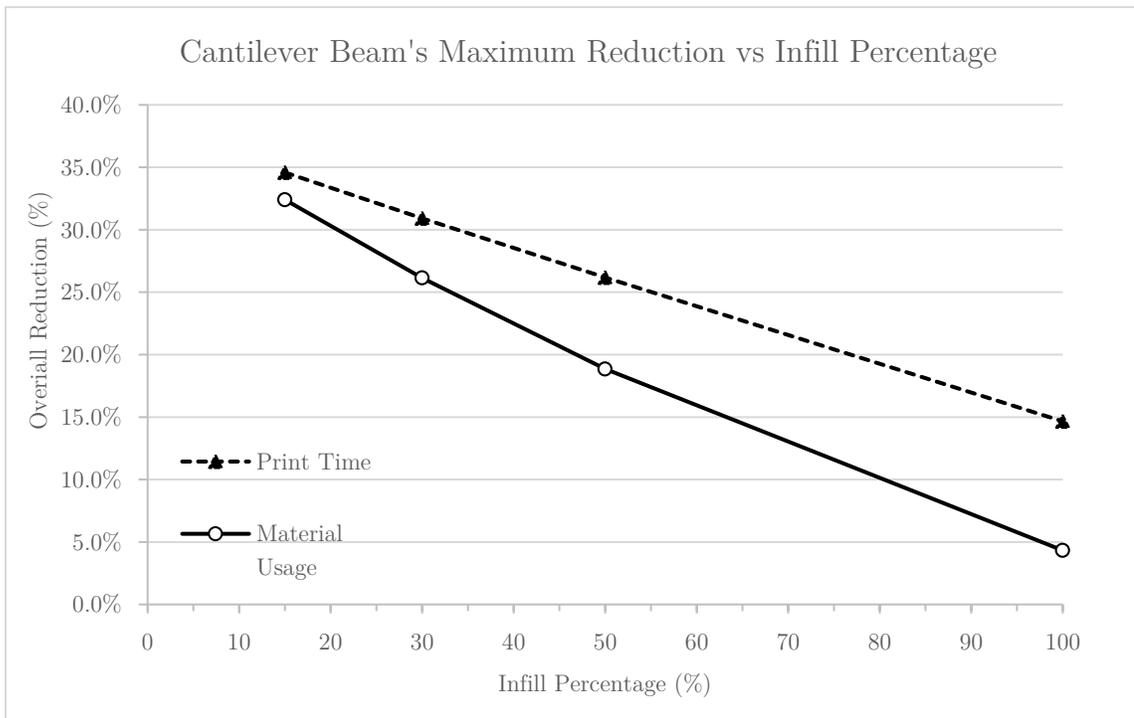
**Figure 5-32.** Resulting toolpaths for various cantilever beam results with identical slicer settings (Slic3r).



**Figure 5-33.** Change in the cantilever beam's simulated print time.



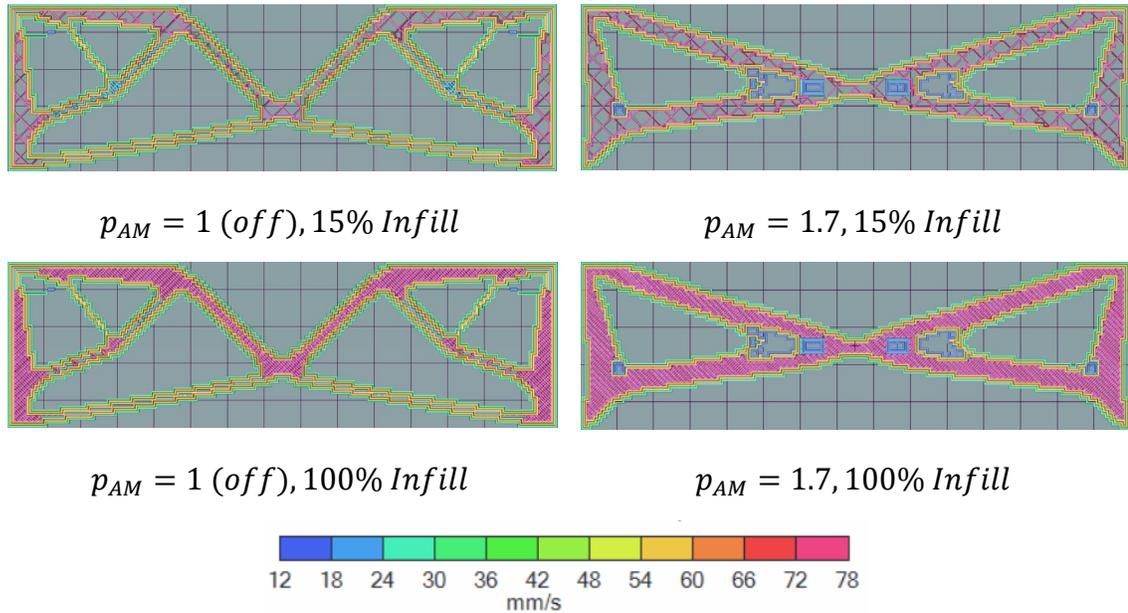
**Figure 5-34.** Change in the cantilever beam's simulated material usage.



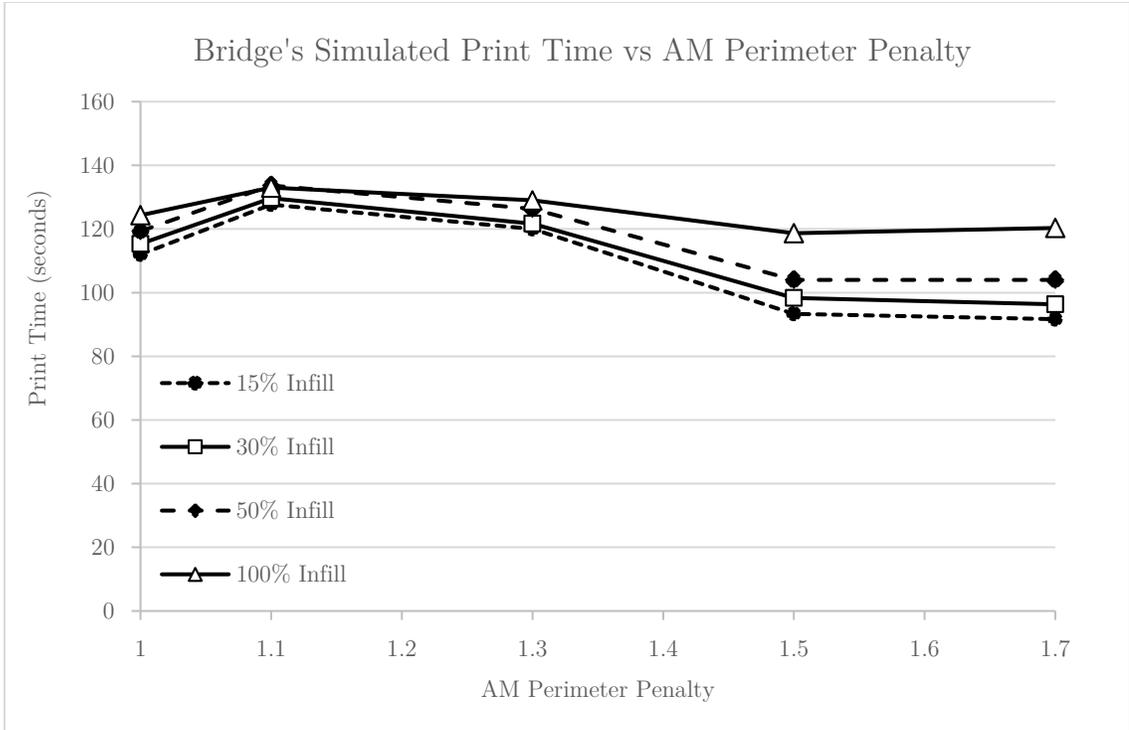
**Figure 5-35.** Change in the cantilever beam's simulated print time and material usage based on the selected infill percentage for  $p_{AM} = 1.7$ .

### 5.3.3 Bridge

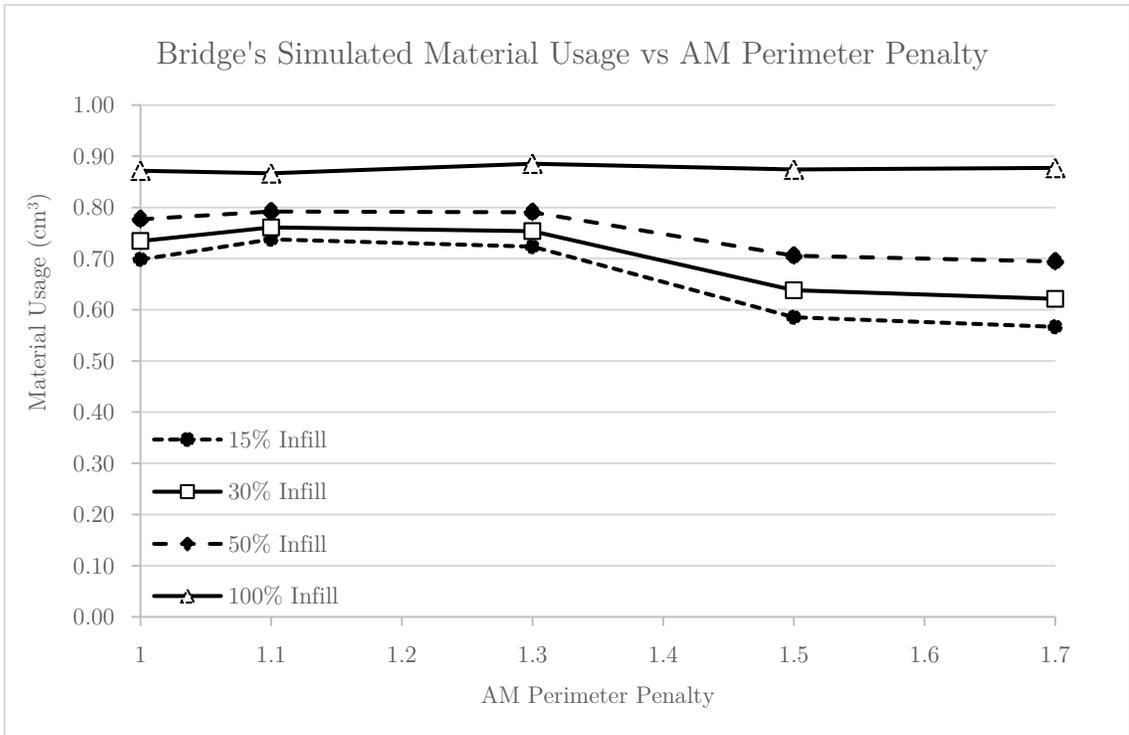
Like the other problems so far, only one layer is tested, with the original geometry being scaled to  $15\text{ cm} \times 4.5\text{ cm}$  to fit on a standard 3D printer bed. A selection of toolpaths are shown in Figure 5-36. The same trends in simulated print times, material usage, and maximum reductions are seen in Figure 5-37, 5-38, and 5-39 respectively. At 15% infill, the maximum print time occurs with a reduction of 18.2%, with the maximum material usage decreasing by 18.9%. The material usage at 100% infill also slightly increases.



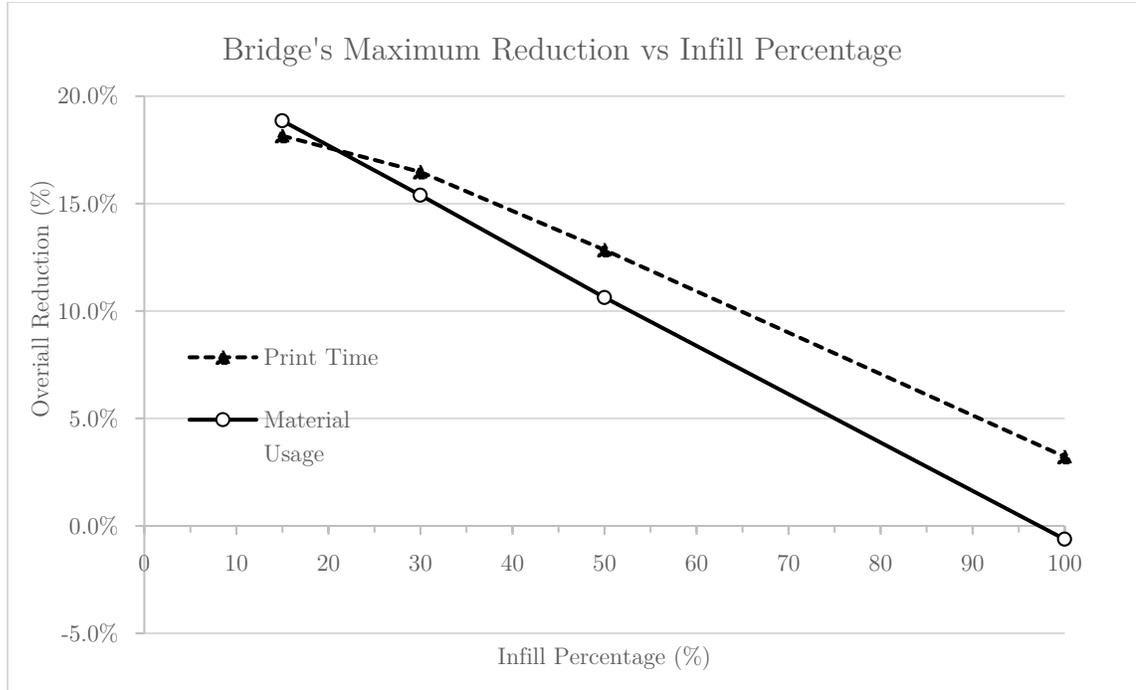
**Figure 5-36.** Resulting toolpaths for various bridge results with identical slicer settings (Slic3r).



**Figure 5-37.** Change in the bridge’s simulated print time.



**Figure 5-38.** Change in the bridge’s simulated material usage.

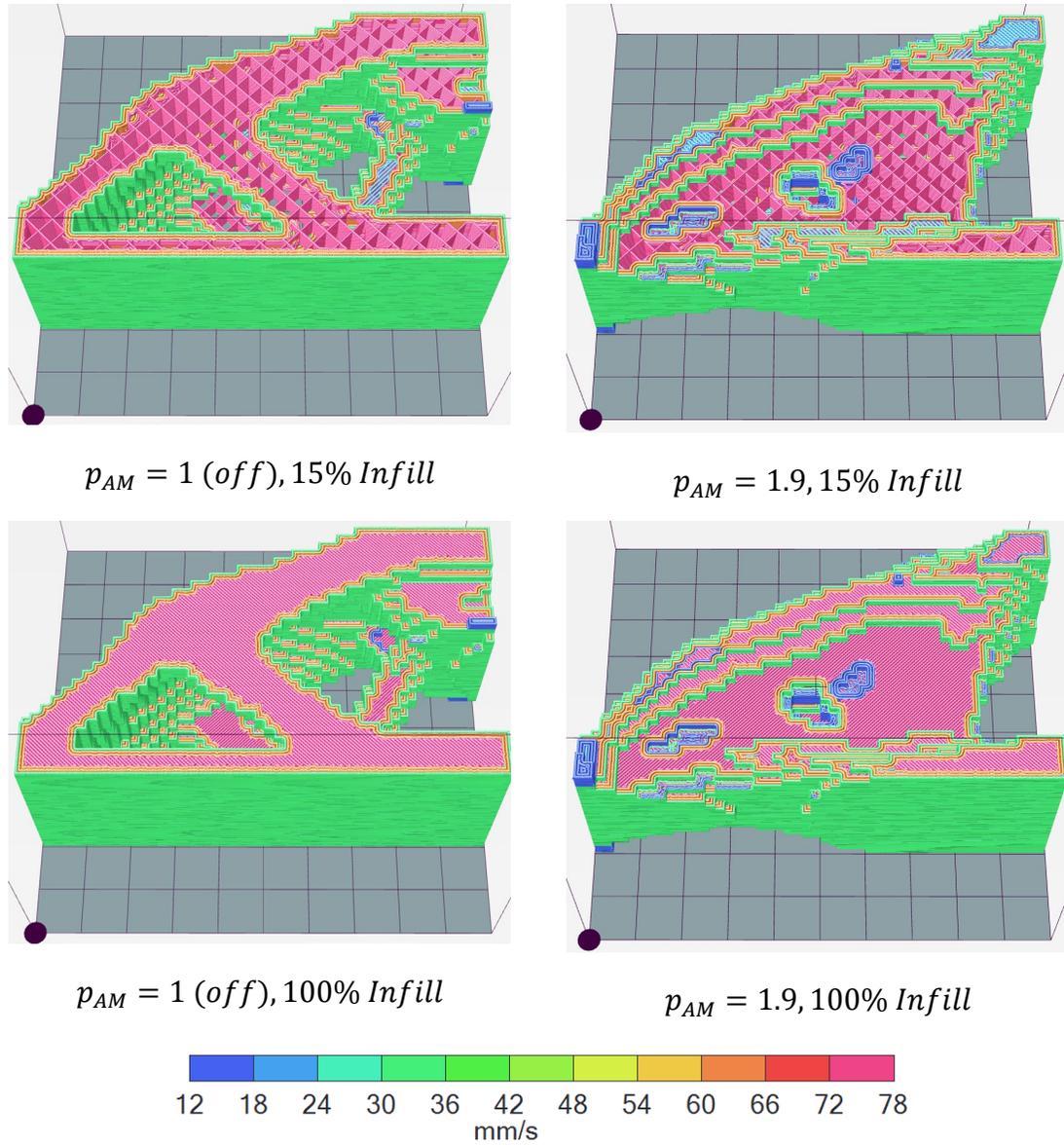


**Figure 5-39.** Change in the bridge’s simulated print time and material usage based on the selected infill percentage for  $p_{AM} = 1.7$ .

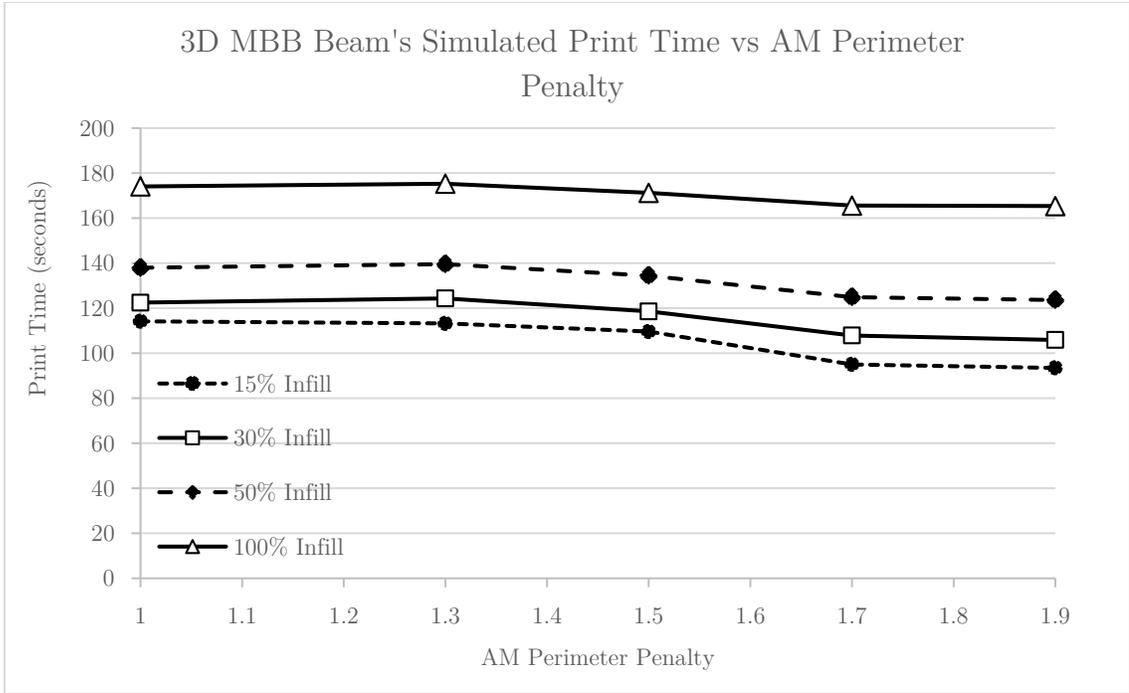
### 5.3.4 3D MBB Beam

The 3D MBB will be scaled to 10 cm × 10 cm. Any smaller and the geometry becomes too small for selected printing parameters to adequately capture the details of the geometry. Due to the 0.3 mm layer thickness selected and the scaling, there will be a total of 133 layers. This means that every 6-7 layers will correspond with elements on that particular slice. Support structures are also omitted in order to isolate the effect of the AM perimeter penalty. Figure 5-40 displays some generated toolpaths. Figures 5-41 and 5-42 detail the simulated print times and material usage for each penalty value. The trends that occurred in the 2D example also apply here. The maximum material and print time reductions are presented in Figure 5-43, with

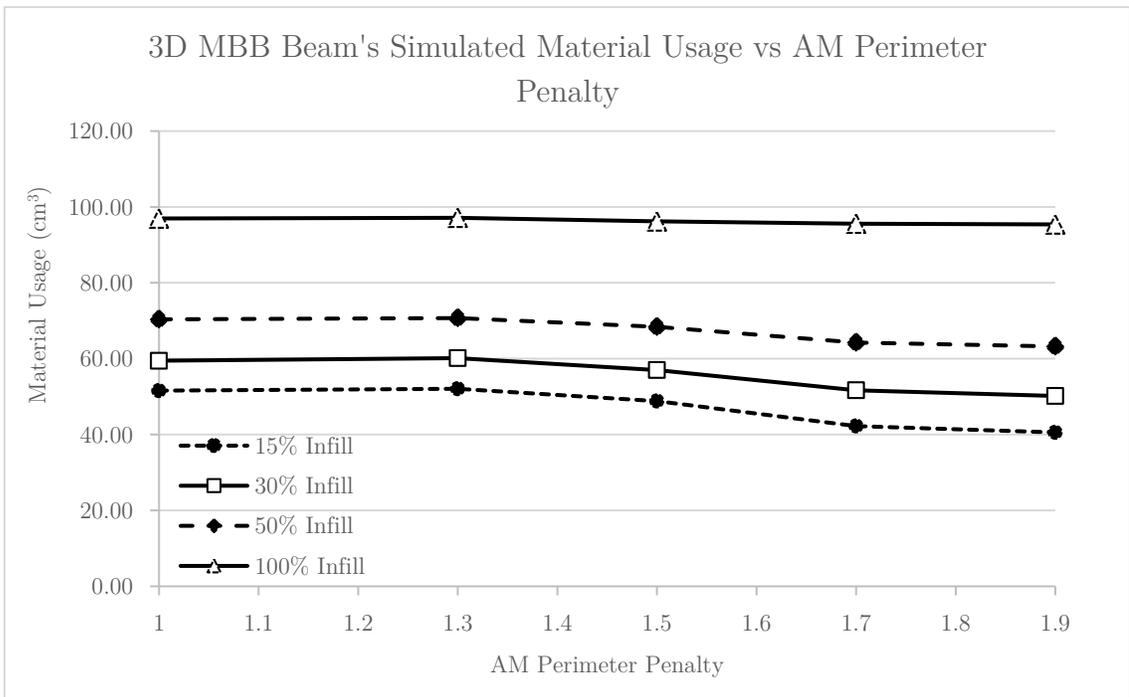
maximum reductions of 18.1% for print time and 21.3% for material usage occurring at 15% infill.



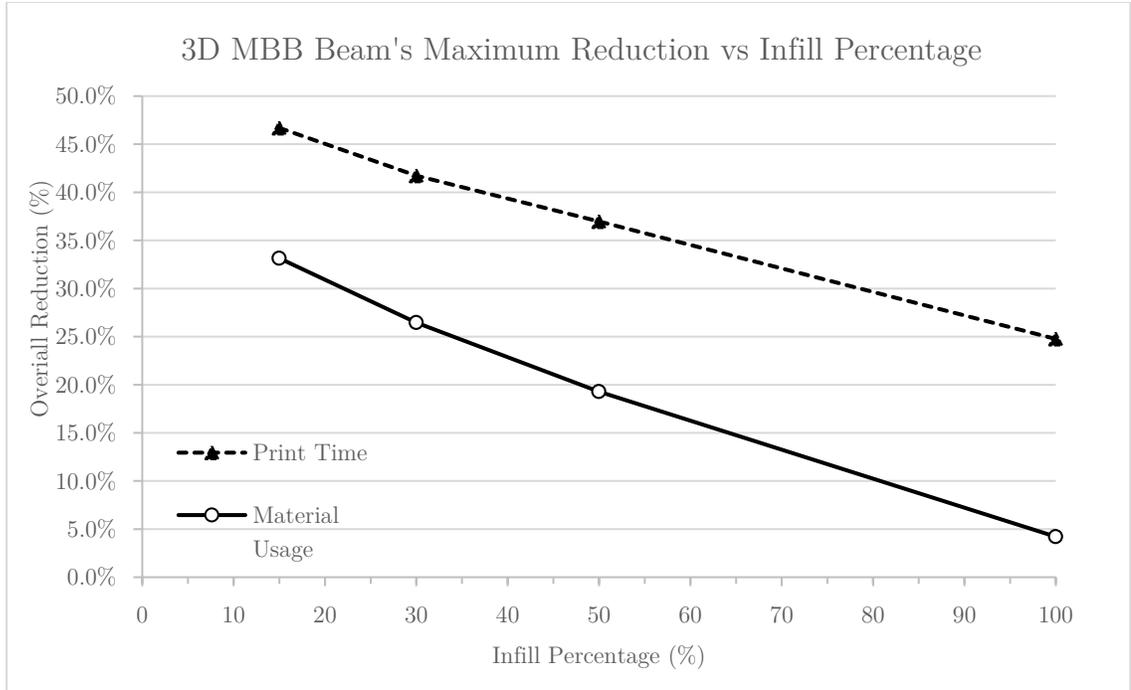
**Figure 5-40.** Resulting toolpaths for various 3D MBB beam results with identical slicer settings (Slic3r).



**Figure 5-41.** Change in the 3D MBB beam's simulated print time.



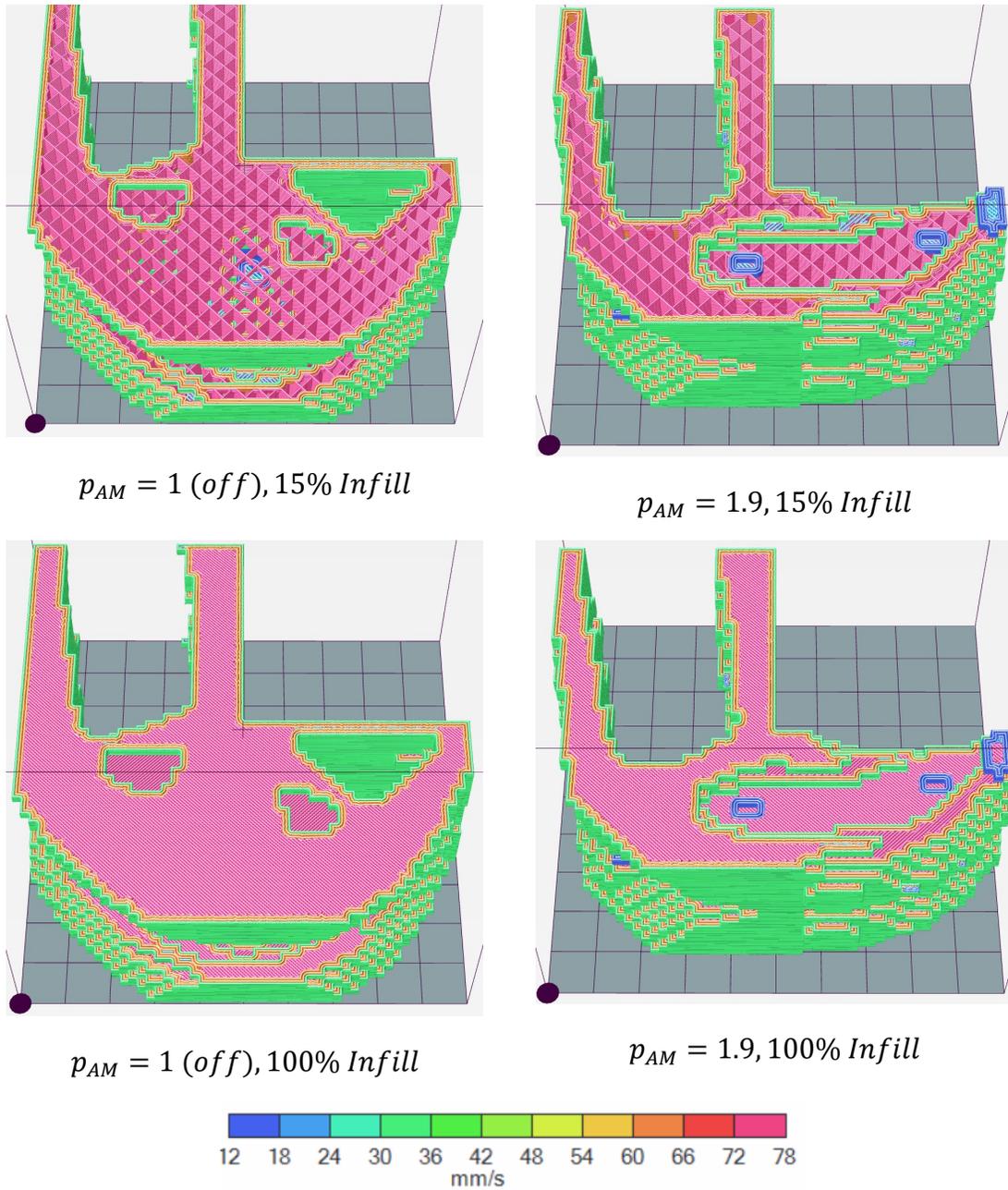
**Figure 5-42.** Change in the 3D MBB beam's simulated material usage.



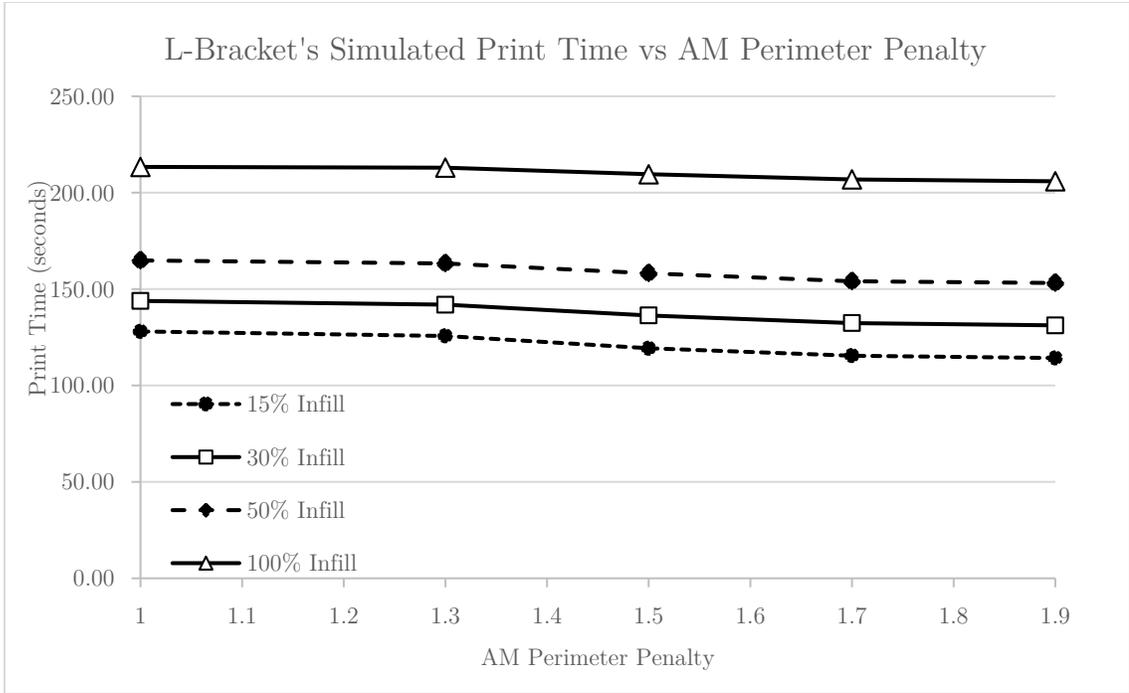
**Figure 5-43.** Change in the 3D MBB beam’s simulated print time and material usage based on the selected infill percentage for  $p_{AM} = 1.9$ .

### 5.3.5 3D L-Bracket

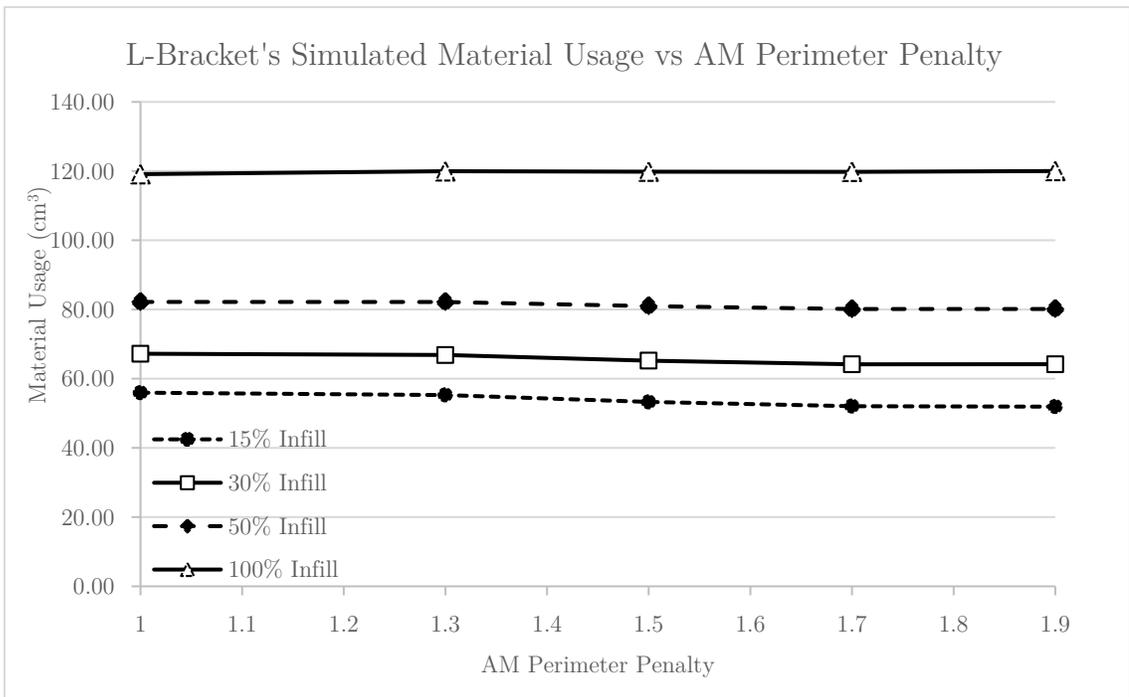
The same scaling will be applied to the 3D L-bracket, fitting it into a 10 cm × 10 cm printing area. Again, there are 133 layers due to this scaling and the 0.3 mm layer thickness. Various toolpaths are presented in Figure 5-44. For each penalty value, the simulated print times and material usage are shown in Figure 5-45 and Figure 5-46. The maximum print time and material gains occur at 15% infill and are 10.8% and 7.2% respectively. While the trends from the previous sections hold here, the gains are significantly reduced in this example. There is also a 0.8% increase in material usage at 100% for the maximum penalty value.



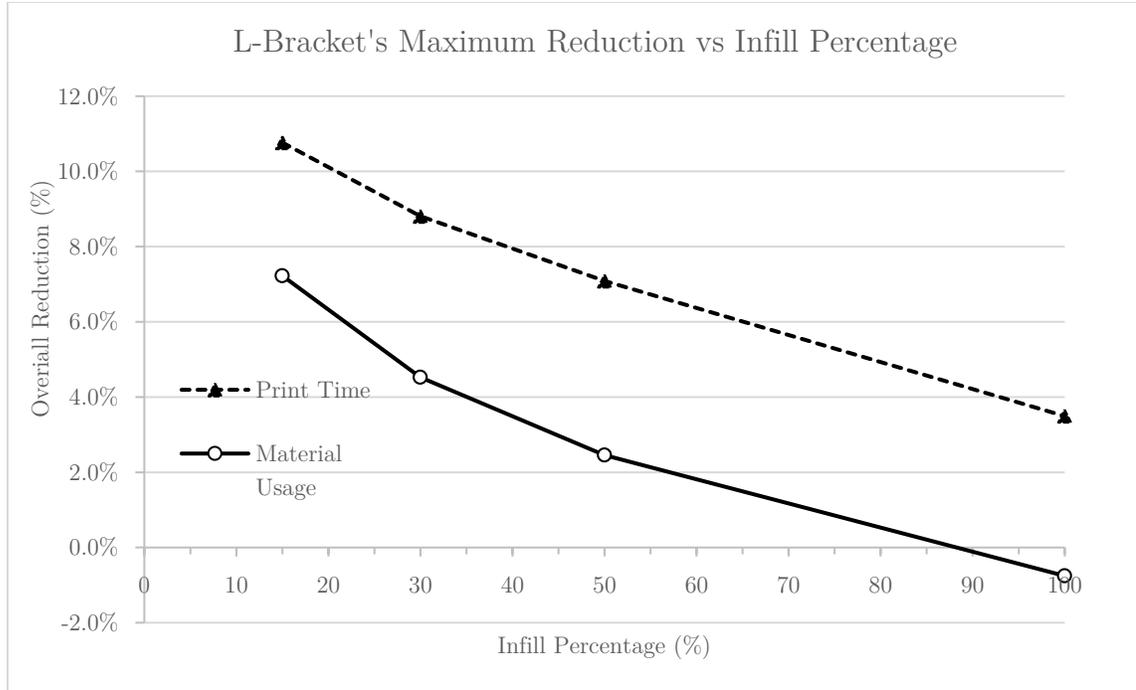
**Figure 5-44.** Resulting toolpaths for various 3D L-bracket results with identical slicer settings (Slic3r).



**Figure 5-45.** Change in the 3D L-bracket's simulated print time.



**Figure 5-46.** Change in the 3D L-bracket's simulated material usage.



**Figure 5-47.** Change in the 3D L-bracket's simulated print time and material usage based on the selected infill percentage for  $p_{AM} = 1.9$ .

## 5.4 Summary of Results

From the presented case examples, it can be seen that the AM topology optimization perimeter constraint results in the reduction of print time. A summary of the results can be seen in Table 5-1. For each case that was presented, the change in compliance, normalized perimeter, print time, and material usage (at 15% infill) are shown for each tested AM perimeter penalty value compared to the results with no penalty,  $p_{AM} = 1$ . It is noted that the maximum reduction in print time and material usage always occurs at the highest penalization value. While the compliance is slightly variable due to the possible presence of local minima (\* specifically noted for the 3D MBB beam example) in the original problem, it generally increases as the penalty value increases.

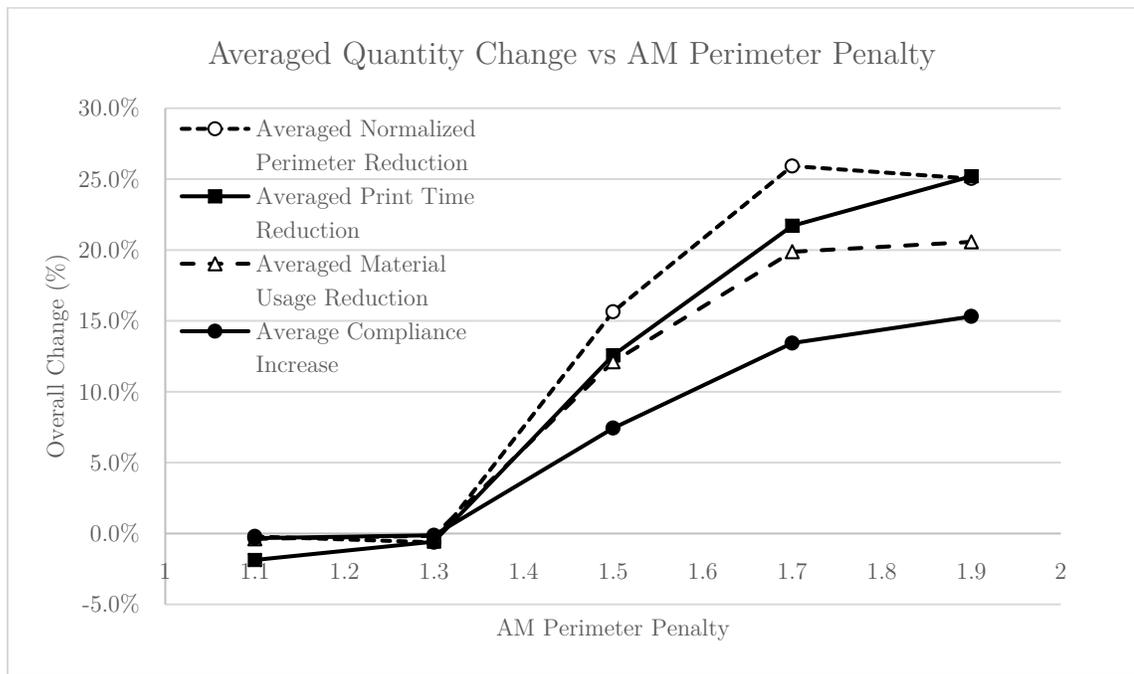
**Table 5-1.** Percentage change of various quantities as compared to the base topology optimization with no AM perimeter penalty.

	$p_{AM} = 1.1$	$p_{AM} = 1.3$	$p_{AM} = 1.5$	$p_{AM} = 1.7$	$p_{AM} = 1.9$
<b>MBB Beam</b>					
<i>Compliance Increase</i>	0.4%	1.9%	18.0%	25.6%	29.3%
<i>Normalized Perimeter Reduction</i>	0.6%	-3.6%	15.1%	30.3%	39.8%
<i>Print Time Reduction</i>	4.3%	1.9%	14.8%	29.2%	46.7%
<i>Material Usage Reduction</i>	1.3%	1.4%	11.4%	23.1%	33.1%
<b>Cantilever Beam</b>					
<i>Compliance Increase</i>	0.0%	-0.9%	18.6%	25.1%	N/A
<i>Normalized Perimeter Reduction</i>	2.4%	-0.3%	31.5%	41.5%	N/A
<i>Print Time Reduction</i>	4.1%	-0.3%	20.5%	34.6%	N/A
<i>Material Usage Reduction</i>	3.2%	1.2%	22.9%	32.4%	N/A
<b>Bridge</b>					
<i>Compliance Increase</i>	-1.4%	-1.3%	3.1%	11.3%	N/A
<i>Normalized Perimeter Reduction</i>	-3.6%	-0.1%	23.9%	30.1%	N/A
<i>Print Time Reduction</i>	-14.0%	-7.1%	16.7%	18.2%	N/A
<i>Material Usage Reduction</i>	-5.6%	-3.6%	16.2%	18.9%	N/A
<b>3D MBB Beam</b>					
<i>Compliance Increase</i>	N/A	-1.1%*	-3.7%*	2.8%*	-0.4%*
<i>Normalized Perimeter Reduction</i>	N/A	-1.2%	3.5%	17.5%	23.6%
<i>Print Time Reduction</i>	N/A	0.8%	4.0%	16.8%	18.1%
<i>Material Usage Reduction</i>	N/A	-0.9%	5.3%	18.1%	21.3%
<b>3D L-Bracket</b>					
<i>Compliance Increase</i>	N/A	0.8%	1.1%	2.3%	16.9%
<i>Normalized Perimeter Reduction</i>	N/A	2.1%	4.1%	10.1%	11.7%
<i>Print Time Reduction</i>	N/A	1.8%	6.8%	9.8%	10.8%
<i>Material Usage Reduction</i>	N/A	1.2%	4.8%	7.0%	7.2%

The addition of greater penalty values also generally increased the number of iterations required to converge the solution, with values higher than what was presented not converging at all. It is speculated that this effect is caused by the prolonged presence of intermediate density material resulting from an over-aggressive perimeter constraint. The final designs also contained some intermediate density

material. To mitigate both of these issues, some form of variable perimeter penalty could be used. It is speculated that a lower perimeter penalty value could be used during the initial stages of the optimization to allow solid members to form quickly. The penalty could then be increased to match the current implementation, and then decreased when the design is almost converged, allowing most intermediate density elements to be removed. A change in the slicer density threshold could also address this.

The averaged change in normalized perimeter, print time, material reduction (at 15% infill), and compliance across all examples is shown in Figure 5-48:



**Figure 5-48.** Change in averaged quantities vs AM perimeter penalty.

Here, a good match between the normalized perimeter and print time can be seen. Penalty values below 1.5 seem to have little effect on the results, with values of 1.7 and 1.9 being most effective. Compliance increase also follows this trend.

# Chapter 6

## Conclusion

A method to converge a structural topology optimization towards a design that reduces additive manufacturing print time was developed and implemented in a new research platform. With this, the presented thesis provides a few contributions to the additive manufacturing topology optimization field. A new approach to reduce print time was detailed using the principle of mass concentration and perimeter penalization. This approach is significant because it recognizes two key features of the AM topology optimization process. First, this method is based on the realization that many 3D printed parts utilize infill to reduce weight and print time. It accounts for this by not trying to reduce the surface area of each layer of the printed part, but rather the layers exterior and interior outlines. These must be deposited to ensure an adequate surface representation. This is also relevant in the context of material distribution schemes where the final surface area of the layers, or rather slices of the part will remain at the constant volume fraction. However, the outline, or perimeters of these slices do not have to be constant, allowing them to be minimized. It is also

key that it differentiates the outer and inner perimeters, as it was shown that adding or removing material on each one results in a different effect.

The second key realization is that this perimeter reduction can be done in the context of the 3D printing process itself. When a part is sent to be printed, it must first be converted to the STL file format, then sliced into cross-sectional layers where toolpaths can be generated for the specific printing process. Here, the presented method slices part geometry to identify the inner and outer contours during each iteration of the topology optimization. It utilizes a state-of-the-art FEA slicer to bypass the need for conversion. Once the contours are identified and categorized, the reduction of perimeter is maximized by penalizing elements lying on the outer contours and giving an advantage to elements on inner contours. The effect is a shrinking of outer perimeters and a collapsing of inner ones. This approach was implemented in commercial FEA code utilizing a separate executable for the topology optimization and slicing scheme. Compared to most research works in this field, this allows for complex problems to be setup easily with the flexibility and robustness of the commercial user interface and solver. The separate scripting executable allows for simple implementation of other research works, with the benefits of a modern, user-friendly coding language.

### **Future Work**

To end, the practical applications and future works of this thesis will be discussed. For the current method and implementation, it is clear that a substantial improvement in print time can be achieved, albeit at the cost of some stiffness. It

also generally reduces the amount of material used in the printing process as a side effect. This method is not particularly useful for a single, one-off part, but rather for a large to medium-scale manufacturing run where even a small reduction in print time or material usage could lead to a more feasible and cost-effective AM based production. It could also be used to allow for a greater proportion of infill for the same printing time. As for future works, several additions and modifications could be made to improve the constraints usefulness. First a parametric study should be done to further quantify the effects of the new constraint, with the slicer density threshold being the main focus. The variable penalty concept could also be explored. Since adaptive slicing is important for maintaining good surface quality while reducing print time, this feature could be implemented to further mirror the true AM printing process. The addition of the overhang constraint to this process would also be extremely relevant, as this work does not consider the additional time required to print support structures. Consideration of the structural performance of infill in the part during the print time minimization would also be highly recommended to ensure this is feature is fully utilized.

# Bibliography

- [1] M. P. Bendsøe and O. Sigmund, *Topology Optimization: Theory, Methods and Applications*. 2003.
- [2] M. P. Bendsøe and N. Kikuchi, “Generating optimal topologies in structural design using a homogenization method,” *Comput. Methods Appl. Mech. Eng.*, vol. 71, no. 2, pp. 197–224, Nov. 1988.
- [3] K.-T. Cheng and N. Olhoff, “An investigation concerning optimal design of solid elastic plates,” *Int. J. Solids Struct.*, vol. 17, no. 3, pp. 305–323, Jan. 1981.
- [4] M. P. Bendsøe, “Optimal shape design as a material distribution problem,” *Struct. Optim.*, vol. 1, no. 4, pp. 193–202, Dec. 1989.
- [5] G. I. N. Rozvany, M. Zhou, and T. Birker, “Generalized shape optimization without homogenization,” *Struct. Optim.*, vol. 4, no. 3–4, pp. 250–252, Sep. 1992.
- [6] M. Zhou and G. I. N. Rozvany, “The COC algorithm, Part II: Topological, geometrical and generalized shape optimization,” *Comput. Methods Appl. Mech. Eng.*, vol. 89, no. 1–3, pp. 309–336, Aug. 1991.
- [7] O. Sigmund, “A 99 line topology optimization code written in Matlab,” *Struct. Multidiscip. Optim.*, vol. 21, no. 2, pp. 120–127, Apr. 2001.
- [8] H. P. Mlejnek, “Some aspects of the genesis of structures,” *Struct. Optim.*, vol. 5, no. 1–2, pp. 64–69, Mar. 1992.
- [9] Y. M. Xie and G. P. Steven, “A simple evolutionary procedure for structural optimization,” *Comput. Struct.*, vol. 49, no. 5, pp. 885–896, Dec. 1993.
- [10] G. Allaire, F. Jouve, and A.-M. Toader, “A level-set method for shape optimization,” *Comptes Rendus Math.*, vol. 334, no. 12, pp. 1125–1130, Jan. 2002.
- [11] G. Allaire, F. Jouve, and A.-M. Toader, “Structural optimization using sensitivity analysis and a level-set method,” *J. Comput. Phys.*, vol. 194, no. 1, pp. 363–393, Feb. 2004.

- [12] M. Y. Wang, X. Wang, and D. Guo, “A level set method for structural topology optimization,” *Comput. Methods Appl. Mech. Eng.*, vol. 192, no. 1–2, pp. 227–246, Jan. 2003.
- [13] J. Sokołowski and A. Żochowski, “Topological derivative for optimal control problems,” *Control Cybern.*, vol. Vol. 28, no 3, pp. 611–625, 1999.
- [14] B. Bourdin and A. Chambolle, “Design-dependent loads in topology optimization,” *ESAIM Control. Optim. Calc. Var.*, vol. 9, pp. 19–48, Jan. 2003.
- [15] O. Sigmund, “On the Design of Compliant Mechanisms Using Topology Optimization,” *Mech. Struct. Mach.*, vol. 25, no. 4, pp. 493–524, Jan. 1997.
- [16] J. Petersson and O. Sigmund, “Slope constrained topology optimization,” *Int. J. Numer. Methods Eng.*, vol. 41, no. 8, pp. 1417–1434, Apr. 1998.
- [17] B. Bourdin, “Filters in topology optimization,” *Int. J. Numer. Methods Eng.*, vol. 50, no. 9, pp. 2143–2158, Mar. 2001.
- [18] T. E. Bruns and D. A. Tortorelli, “Topology optimization of non-linear elastic structures and compliant mechanisms,” *Comput. Methods Appl. Mech. Eng.*, vol. 190, no. 26–27, pp. 3443–3459, Mar. 2001.
- [19] J. K. Guest, J. H. Prévost, and T. Belytschko, “Achieving minimum length scale in topology optimization using nodal design variables and projection functions,” *Int. J. Numer. Methods Eng.*, vol. 61, no. 2, pp. 238–254, Sep. 2004.
- [20] J. K. Guest, “Topology optimization with multiple phase projection,” *Comput. Methods Appl. Mech. Eng.*, vol. 199, no. 1–4, pp. 123–135, Dec. 2009.
- [21] W. Zhang, W. Zhong, and X. Guo, “An explicit length scale control approach in SIMP-based topology optimization,” *Comput. Methods Appl. Mech. Eng.*, vol. 282, pp. 71–86, Dec. 2014.
- [22] M. Zhou, B. S. Lazarov, F. Wang, and O. Sigmund, “Minimum length scale in topology optimization by geometric constraints,” *Comput. Methods Appl. Mech. Eng.*, vol. 293, pp. 266–282, Aug. 2015.
- [23] M. Zhou, Y. K. Shyy, and H. L. Thomas, “Checkerboard and minimum member size control in topology optimization,” *Struct. Multidiscip. Optim.*, vol. 21, no. 2, pp. 152–158, Apr. 2001.
- [24] M. Zhou, R. Fleury, Y.-K. Shyy, H. Thomas, and J. Brennan, “Progress in Topology Optimization with Manufacturing Constraints,” in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002.
- [25] U. Schramm and M. Zhou, “Recent Developments in the Commercial Implementation of Topology Optimization,” in *IUTAM Symposium on*

- Topological Design Optimization of Structures, Machines and Materials*, Springer Netherlands, 2006, pp. 239–248.
- [26] S. Liu, Q. Li, W. Chen, L. Tong, and G. Cheng, “An identification method for enclosed voids restriction in manufacturability design for additive manufacturing structures,” *Front. Mech. Eng.*, vol. 10, no. 2, pp. 126–137, Jun. 2015.
- [27] Q. Li, W. Chen, S. Liu, and L. Tong, “Structural topology optimization considering connectivity constraint,” *Struct. Multidiscip. Optim.*, vol. 54, no. 4, pp. 971–984, Oct. 2016.
- [28] A. Lalehpour and A. Barari, “Post processing for Fused Deposition Modeling Parts with Acetone Vapour Bath,” in *12th IFAC Workshop on Intelligent Manufacturing Systems IMS 2016*, 2016, vol. 49, no. 31, pp. 42–48.
- [29] S. Allen and D. Dutta, “On the computation of part orientation using support structures in layered manufacturing,” in *1994 International Solid Freeform Fabrication Symposium*, 1994, pp. 259–269.
- [30] H. D. Morgan, J. A. Cherry, S. Jonnalagadda, D. Ewing, and J. Sienz, “Part orientation optimisation for the additive layer manufacture of metal components,” *Int. J. Adv. Manuf. Technol.*, vol. 86, no. 5–8, pp. 1679–1687, Sep. 2016.
- [31] X. Zhang, X. Le, A. Panotopoulou, E. Whiting, and C. C. L. Wang, “Perceptual models of preference in 3D printing direction,” *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–12, Oct. 2015.
- [32] X. Huang, C. Ye, S. Wu, K. Guo, and J. Mo, “Sloping wall structure support generation for fused deposition modeling,” *Int. J. Adv. Manuf. Technol.*, vol. 42, no. 11–12, pp. 1074–1081, Jun. 2009.
- [33] J. Vanek, J. A. G. Galicia, and B. Benes, “Clever Support: Efficient Support Structure Generation for Digital Fabrication,” *Comput. Graph. Forum*, vol. 33, no. 5, pp. 117–125, Aug. 2014.
- [34] M. X. Gan and C. H. Wong, “Practical support structures for selective laser melting,” *J. Mater. Process. Technol.*, vol. 238, pp. 474–484, Dec. 2016.
- [35] J. Dumas, J. Hergel, and S. Lefebvre, “Bridging the gap: automated steady scaffoldings for 3D printing,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–10, Jul. 2014.
- [36] G. Strano, L. Hao, R. M. Everson, and K. E. Evans, “A new approach to the design and optimisation of support structures in additive manufacturing,” *Int. J. Adv. Manuf. Technol.*, vol. 66, no. 9–12, pp. 1247–1254, Jun. 2013.
- [37] D. Brackett, I. Ashcroft, and R. Hague, “Topology optimization for additive manufacturing,” in *Proceedings of the solid freeform fabrication symposium*,

- 2011, pp. 348–362.
- [38] M. Leary, L. Merli, F. Torti, M. Mazur, and M. Brandt, “Optimal topology for additive manufacture: A method for enabling additive manufacture of support-free optimal structures,” *Mater. Des.*, vol. 63, pp. 678–690, Nov. 2014.
- [39] A. T. Gaynor and J. K. Guest, “Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design,” *Struct. Multidiscip. Optim.*, vol. 54, no. 5, pp. 1157–1172, Nov. 2016.
- [40] A. T. Gaynor, N. A. Meisel, C. B. Williams, and J. K. Guest, “Topology Optimization for Additive Manufacturing: Considering Maximum Overhang Constraint,” in *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2014.
- [41] M. Langelaar, “Topology optimization of 3D self-supporting structures for additive manufacturing,” *Addit. Manuf.*, vol. 12, pp. 60–70, Oct. 2016.
- [42] M. Langelaar, “An additive manufacturing filter for topology optimization of print-ready designs,” *Struct. Multidiscip. Optim.*, vol. 55, no. 3, pp. 871–883, Mar. 2017.
- [43] X. Qian, “Undercut and overhang angle control in topology optimization: A density gradient based integral approach,” *Int. J. Numer. Methods Eng.*, vol. 111, no. 3, pp. 247–272, Jul. 2017.
- [44] A. M. Mirzendehtel and K. Suresh, “Support structure constrained topology optimization for additive manufacturing,” *Comput. Des.*, vol. 81, pp. 1–13, Dec. 2016.
- [45] D. Jankovics, H. Gohari, M. Tayefeh, and A. Barari, “Developing Topology Optimization with Additive Manufacturing Constraints in ANSYS®,” in *16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018*, 2018, vol. 51, no. 11, pp. 1359–1364.
- [46] D. Jankovics, H. Gohari, and A. Barari, “Constrained Topology Optimization For Additive Manufacturing Of Structural Components In Ansys®,” in *Proceedings of The Canadian Society for Mechanical Engineering International Congress 2018*, 2018.
- [47] Y. Mass and O. Amir, “Topology optimization for additive manufacturing: Accounting for overhang limitations using a virtual skeleton,” *Addit. Manuf.*, vol. 18, pp. 58–73, Dec. 2017.
- [48] Y. Mass and O. Amir, “Using a virtual skeleton to increase printability of topology optimized design for industry-class applications,” *Comptes Rendus Mécanique*, vol. 346, no. 11, pp. 1104–1121, Nov. 2018.
- [49] A. M. Driessen, “Overhang constraint in topology optimisation for additive

- manufacturing: A density gradient based approach,” TU Delft, 2016.
- [50] M. Langelaar, “Combined optimization of part topology, support structure layout and build orientation for additive manufacturing,” *Struct. Multidiscip. Optim.*, pp. 1–20, 2018.
- [51] A. Dolenc and I. Mäkelä, “Slicing procedures for layered manufacturing techniques,” *Comput. Des.*, vol. 26, no. 2, pp. 119–126, Feb. 1994.
- [52] E. Sabourin, S. A. Houser, and J. Helge Bøhn, “Adaptive slicing using stepwise uniform refinement,” *Rapid Prototyp. J.*, vol. 2, no. 4, pp. 20–26, Dec. 1996.
- [53] J. Tyberg and J. Helge Bøhn, “Local adaptive slicing,” *Rapid Prototyp. J.*, vol. 4, no. 3, pp. 118–127, Sep. 1998.
- [54] D. Cormier, K. Unnanon, and E. Sanii, “Specifying non-uniform cusp heights as a potential aid for adaptive slicing,” *Rapid Prototyp. J.*, vol. 6, no. 3, pp. 204–212, Sep. 2000.
- [55] R. Jamieson and H. Hacker, “Direct slicing of CAD models for rapid prototyping,” *Rapid Prototyp. J.*, vol. 1, no. 2, pp. 4–12, Jun. 1995.
- [56] Z. Zhao and L. Laperrière, “Adaptive direct slicing of the solid model for rapid prototyping,” *Int. J. Prod. Res.*, vol. 38, no. 1, pp. 69–83, Jan. 2000.
- [57] S. Sikder, A. Barari, and H. A. Kishawy, “Global adaptive slicing of NURBS based sculptured surface for minimum texture error in rapid prototyping,” *Rapid Prototyp. J.*, vol. 21, no. 6, pp. 649–661, Oct. 2015.
- [58] A. Barari, H. A. Kishawy, F. Kaji, and M. A. Elbestawi, “On the surface quality of additive manufactured parts,” *Int. J. Adv. Manuf. Technol.*, vol. 89, no. 5–8, pp. 1969–1974, Mar. 2017.
- [59] H. Gohari, A. Barari, and H. Kishawy, “Using Multistep Methods in Slicing 2 ½ Dimensional Parametric Surfaces for Additive Manufacturing Applications,” in *12th IFAC Workshop on Intelligent Manufacturing Systems IMS 2016*, 2016, vol. 49, no. 31, pp. 67–72.
- [60] J. Liu, Y. Ma, A. J. Qureshi, and R. Ahmad, “Light-weight shape and topology optimization with hybrid deposition path planning for FDM parts,” *Int. J. Adv. Manuf. Technol.*, vol. 97, no. 1–4, pp. 1123–1135, Jul. 2018.
- [61] D. Bender and A. Barari, “Direct Solid Element Slicing for Additive Manufacturing,” in *Proceedings of the ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, (Pre-print)*, 2019.
- [62] S. Ahn, M. Montero, D. Odell, S. Roundy, and P. K. Wright, “Anisotropic material properties of fused deposition modeling ABS,” *Rapid Prototyp. J.*, vol. 8, no. 4, pp. 248–257, Oct. 2002.
- [63] N. Umetani and R. Schmidt, “Cross-sectional structural analysis for 3D

- printing optimization,” in *SIGGRAPH Asia Technical Briefs*, 2013, pp. 5–1.
- [64] J. Liu, “Guidelines for AM part consolidation,” *Virtual Phys. Prototyp.*, vol. 11, no. 2, pp. 133–141, Apr. 2016.
- [65] A. M. Mirzendehtel, B. Rankouhi, and K. Suresh, “Strength-based topology optimization for anisotropic parts,” *Addit. Manuf.*, vol. 19, pp. 104–113, Jan. 2018.
- [66] R. Hoglund and D. Smith, “Continuous Fiber Angle Topology Optimization for Polymer Fused Filament Fabrication,” in *Proceedings of the 27th Annual International Solid Freeform Fabrication Symposium*, 2016.
- [67] D. Jiang, R. Hoglund, D. Smith, D. Jiang, R. Hoglund, and D. E. Smith, “Continuous Fiber Angle Topology Optimization for Polymer Composite Deposition Additive Manufacturing Applications,” *Fibers*, vol. 7, no. 2, p. 14, Feb. 2019.
- [68] J. Liu and H. Yu, “Concurrent deposition path planning and structural topology optimization for additive manufacturing,” *Rapid Prototyp. J.*, vol. 23, no. 5, pp. 930–942, Aug. 2017.
- [69] J. Liu, Y. Zheng, Y. Ma, A. Qureshi, and R. Ahmad, “A Topology Optimization Method for Hybrid Subtractive–Additive Remanufacturing,” *Int. J. Precis. Eng. Manuf. Technol.*, pp. 1–15, Mar. 2019.
- [70] C. Dapogny, R. Estevez, A. Faure, and G. Michailidis, “Shape and topology optimization considering anisotropic features induced by additive manufacturing processes,” *Comput. Methods Appl. Mech. Eng.*, vol. 344, pp. 626–665, Feb. 2019.
- [71] E. Sabourin, S. A. Houser, and J. Helge Bøhn, “Accurate exterior, fast interior layered manufacturing,” *Rapid Prototyp. J.*, vol. 3, no. 2, pp. 44–52, Jun. 1997.
- [72] O. Sigmund, S. Torquato, and I. A. Aksay, “On the design of 1–3 piezocomposites using topology optimization,” *J. Mater. Res.*, vol. 13, no. 4, pp. 1038–1048, Apr. 1998.
- [73] O. Sigmund, “A new class of extremal composites,” *J. Mech. Phys. Solids*, vol. 48, no. 2, pp. 397–428, Feb. 2000.
- [74] H. Rodrigues, J. M. Guedes, and M. P. Bendsoe, “Hierarchical optimization of material and structure,” *Struct. Multidiscip. Optim.*, vol. 24, no. 1, pp. 1–10, Aug. 2002.
- [75] S. J. Hollister, “Porous scaffold design for tissue engineering,” *Nat. Mater.*, vol. 4, no. 7, pp. 518–524, Jul. 2005.
- [76] P. G. Coelho, P. R. Fernandes, J. M. Guedes, and H. C. Rodrigues, “A hierarchical model for concurrent material and topology optimisation of three-dimensional structures,” *Struct. Multidiscip. Optim.*, vol. 35, no. 2, pp. 107–

- 115, Feb. 2008.
- [77] P. G. Coelho, J. M. Guedes, and H. C. Rodrigues, “Multiscale topology optimization of bi-material laminated composite structures,” *Compos. Struct.*, vol. 132, pp. 495–505, Nov. 2015.
- [78] X. Zhang, Y. Xia, J. Wang, Z. Yang, C. Tu, and W. Wang, “Medial axis tree—an internal supporting structure for 3D printing,” *Comput. Aided Geom. Des.*, vol. 35–36, pp. 149–162, May 2015.
- [79] J. Wu, C. C. L. Wang, X. Zhang, and R. Westermann, “Self-supporting rhombic infill structures for additive manufacturing,” *Comput. Des.*, vol. 80, pp. 32–42, Nov. 2016.
- [80] J. Wu, N. Aage, R. Westermann, and O. Sigmund, “Infill Optimization for Additive Manufacturing—Approaching Bone-Like Porous Structures,” *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 2, pp. 1127–1140, Feb. 2018.
- [81] C. Liu, Z. Du, W. Zhang, Y. Zhu, and X. Guo, “Additive Manufacturing-Oriented Design of Graded Lattice Structures Through Explicit Topology Optimization,” *J. Appl. Mech.*, vol. 84, no. 8, p. 081008, Jun. 2017.
- [82] J. Wu, A. Clausen, and O. Sigmund, “Minimum compliance topology optimization of shell–infill composites for additive manufacturing,” *Comput. Methods Appl. Mech. Eng.*, vol. 326, pp. 358–375, Nov. 2017.
- [83] A. Clausen, E. Andreassen, and O. Sigmund, “Topology optimization of 3D shell structures with porous infill,” *Acta Mech. Sin.*, vol. 33, no. 4, pp. 778–791, Aug. 2017.
- [84] R. Huang, E. Ulu, L. Kara, and K. Whitefoot, “Cost minimization in metal additive manufacturing using concurrent structure and process optimization,” in *Proceedings of the ASME 2017 International Design Engineering Technical Conference & Computers and Information in Engineering Conference*, 2017.
- [85] J. Liu, Q. Chen, X. Liang, and A. C. To, “Manufacturing cost constrained topology optimization for additive manufacturing,” *Front. Mech. Eng.*, vol. 14, no. 2, pp. 213–221, Jun. 2019.
- [86] D. Jankovics and A. Barari, “Customization of Automotive Structural Components using Additive Manufacturing and Topology Optimization,” in *13th IFAC Workshop on Intelligent Manufacturing Systems, (Pre-print)*, 2019.
- [87] G. Sabiston, L. Ryan, and I. Y. Kim, “Minimization of Cost and Print Time of Additive Manufacturing via Topology Optimization,” in *Volume 1A: 38th Computers and Information in Engineering Conference*, 2018, p. V01AT02A040.
- [88] L. Ryan and I. Y. Kim, “A multiobjective topology optimization approach for cost and time minimization in additive manufacturing,” *Int. J. Numer.*

## BIBLIOGRAPHY

- Methods Eng.*, vol. 118, no. 7, pp. 371–394, May 2019.
- [89] Vergil (Publius Vergilius Maro), *Aeneid I*. 19BC.
- [90] J. Steiner, “Einfache Beweise der isoperimetrischen Hauptsätze,” *J. für die reine und Angew. Math.*, vol. 18, pp. 281–296, 1838.
- [91] A. Siegel, “A Historical Review of the Isoperimetric Theorem in 2-D, and its place in Elementary Plane Geometry,” New York, 2003.
- [92] O. Sigmund, “A 99 line topology optimization code written in Matlab,” *Struct. Multidiscip. Optim.*, vol. 21, no. 2, pp. 120–127, Apr. 2001.